Reward-Free RL is No Harder Than Reward-Aware RL in Linear Markov Decision Processes

Andrew Wagenmaker [1] **Yifang Chen** [1] **Max Simchowitz** [2] **Simon S. Du** [1] **Kevin Jamieson** [1]

Abstract

Reward-free reinforcement learning (RL) considers the setting where the agent does not have access to a reward function during exploration, but must propose a near-optimal policy for an arbitrary reward function revealed only after exploring. In the the tabular setting, it is well known that this is a more difficult problem than reward-aware (PAC) RL—where the agent has access to the reward function during exploration—with optimal sample complexities in the two settings differing by a factor of $|\mathcal{S}|$, the size of the state space. We show that this separation does not exist in the setting of linear MDPs. We first develop a computationally efficient algorithm for reward-free RL in a d-dimensional linear MDP with sample complexity scaling as $\widetilde{\mathcal{O}}(d^2H^5/\epsilon^2)$. We then show a lower bound with matching dimension-dependence of $\Omega(d^2H^2/\epsilon^2)$, which holds for the reward-aware RL setting. To our knowledge, our approach is the first computationally efficient algorithm to achieve optimal d dependence in linear MDPs, even in the single-reward PAC setting. Our algorithm relies on a novel procedure which efficiently traverses a linear MDP, collecting samples in any given "feature direction", and enjoys a sample complexity scaling optimally in the (linear MDP equivalent of the) maximal state visitation probability. We show that this exploration procedure can also be applied to solve the problem of obtaining "well-conditioned" covariates in linear MDPs.

[1]Paul G. Allen School of Computer Science and Engineering, University of Washington, Seattle [2]CSAIL, MIT, Cambridge, MA. Correspondence to: Andrew Wagenmaker <ajwagen@cs.washington.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

1. Introduction

Efficiently exploring unknown, stochastic environments is a central challenge in online reinforcement learning (RL). Whether attempting to learn a near-optimal policy or achieving large online reward, virtually all provably efficient algorithms rely on some form of exploration to guarantee the state space of the Markov Decision Process (MDP) is traversed. Indeed, failure to guarantee such exploration can result in very suboptimal performance, as near-optimal actions may be under-explored, leading to missed reward.

The *reward-free* RL setting highlights the role of exploration in RL by tasking the learner with exploring their environment without access to a reward function, then revealing a reward function, and asking the learner to produce a near-optimal policy for that reward function. In a sense, to solve this problem the learner must traverse the *entire* MDP, as they do not know which states and actions will lead to high reward under the yet-to-be-revealed reward function. In contrast, the *reward-aware* RL setting (also known as the PAC RL setting) gives the learner access to the reward function from the beginning. The learner must again explore the MDP so as to learn a near-optimal policy, but now they may direct their exploration to focus, for instance, on the regions of the environment with large reward.

In the setting of tabular MDPs, MDPs with a finite number of states and actions, it is well known that the reward-free problem is more difficult than the reward-aware problem. Indeed, for an MDP with $|\mathcal{S}|$ states and $|\mathcal{A}|$ actions, to find an ϵ-optimal policy, the sample complexity in the reward-free setting is known to scale as (ignoring H factors) $\Theta(|\mathcal{S}|^2|\mathcal{A}|/\epsilon^2)$, while in the reward-aware setting it scales as $\Theta(|\mathcal{S}||\mathcal{A}|/\epsilon^2)$, a difference of $|\mathcal{S}|$. Intuitively, this reduction in complexity results from the above observation: when given access to a reward function, the learner need not learn every transition to equal precision, but can focus on the most relevant ones, those leading to high reward.

Recently, the RL community has turned its attention to MDPs with large state-spaces, showing that efficient learning is possible with function approximation techniques. While much progress has been made, fundamental questions remain: the optimal rates are not known for either the

reward-free RL or reward-aware RL problems, and it is also not known whether the rates exhibit a gap similar to the one in the tabular setting.

In this work we study *linear function approximation*, in particular the linear MDP setting, and show that, up to H factors, ***reward-free RL is no harder than reward-aware RL in linear MDPs***. We develop a computationally efficient reward-free algorithm which, in a d-dimensional linear MDP, is able to learn an ϵ-optimal policy for an arbitrary number of reward function using only $\tilde{\mathcal{O}}(d^2H^5/\epsilon^2)$ samples. We then show a lower bound on reward-aware RL of $\Omega(d^2H^2/\epsilon^2)$. Our results imply that in RL with function approximation, there is no advantage to "directing" exploration given access to a reward function: in the worst case, all transitions must still be explored. Furthermore, to our knowledge, this is the first result that settles the optimal d dependence attainable by a computationally efficient algorithm in a linear MDP in any problem setting (reward-free, reward-aware, or regret minimization).

Our results critically rely on a novel exploration strategy able to generate "covering trajectories". In particular, our procedure is able to traverse the MDP and collect data in each feature direction up to a given, desired tolerance. Critically, the complexity of this procedure scales with the inverse of the "set visitation" probability, rather than the inverse squared, which proves essential in obtaining the optimal sample complexity. As a corollary of this approach, we also show how to collect well-conditioned covariates: covariates with lower-bounded minimum eigenvalue.

2. Related Work

We highlight three directions in RL that our work relates to.

Exploration in Reinforcement Learning. Arguably the most common approach to exploration in RL is that of *optimism*, which is used by a host of works (Azar et al., 2017; Jin et al., 2018; Zanette & Brunskill, 2019; Jin et al., 2020b; Zhou et al., 2020; Zhang et al., 2020b), and is typically employed to balance the exploration-exploitation tradeoff and achieve low regret. The *reward-free* RL setting seeks to explore an MDP without access to a reward function, in order to then determine a near-optimal policy for an arbitrary reward function. This has been studied in the tabular setting (Jin et al., 2020a; Ménard et al., 2020; Zhang et al., 2020a; Wu et al., 2021), where the optimal scaling is known to be $\Theta(|\mathcal{S}|^2|\mathcal{A}|/\epsilon^2)$ (Jin et al., 2020a), as well as the function approximation setting (Zanette et al., 2020c; Wang et al., 2020; Zhang et al., 2021a). In the linear MDP setting, Wang et al. (2020) show a sample complexity of $\mathcal{O}(\frac{d^3H^6}{\epsilon^2})$, and Zanette et al. (2020c) show a complexity of $\mathcal{O}(\frac{d^3H^5}{\epsilon^2})$, yet neither provides lower bounds. A related line of work on "reward-free" RL in linear MDPs seeks to learn a good feature representation (Agarwal et al., 2020; Modi et al., 2021). The exploration strategy we employ is related to that proposed in Zhang et al. (2020a) and extended by Wagenmaker et al. (2021b), though both of these works consider the tabular setting. A final work of note is Tarbouriech et al. (2020), which seeks to collect an arbitrary desired number of samples from each state in a tabular setting.

Reinforcement Learning with Function Approximation. A subject of much recent interest in the RL community is that of RL with function approximation. The majority of attention has been devoted to MDPs with linear structure (Yang & Wang, 2019; Jin et al., 2020b; Wang et al., 2019; Du et al., 2019; Zanette et al., 2020a;b; Ayoub et al., 2020; Jia et al., 2020; Modi et al., 2020; Weisz et al., 2021; Zhou et al., 2020; 2021; Zhang et al., 2021b; Wang et al., 2021; Wagenmaker et al., 2021a; Huang et al., 2022). Several different settings of MDPs with linear structure have been proposed. Most common are the linear MDP assumption (Jin et al., 2020b), which is the setting we consider in this work, and the linear mixture MDP assumption (Jia et al., 2020; Ayoub et al., 2020; Zhou et al., 2020). A second line of work seeks to determine more general conditions that allow for efficient learning in large state-space MDPs and general function approximation techniques (Jiang et al., 2017; Du et al., 2021; Jin et al., 2021; Foster et al., 2021).

Reward-Aware Reinforcement Learning. Reward-aware RL (also known as PAC RL) has a long history in reinforcement learning, in particular in the tabular setting (Kearns & Singh, 2002; Kakade, 2003; Dann & Brunskill, 2015; Dann et al., 2017). The current state of the art in tabular PAC RL, which achieves an optimal scaling of $\mathcal{O}(|\mathcal{S}||\mathcal{A}|/\epsilon^2)$, is (Dann et al., 2019; Ménard et al., 2020). In the linear MDP setting, the literature has tended to focus on achieving low regret. However, any low-regret algorithm can be used to obtain an ϵ-optimal policy, thereby solving the PAC problem, via an *online-to-batch conversion* (Jin et al., 2018; Ménard et al., 2020). Existing low-regret algorithms in linear MDPs (Jin et al., 2020b; Zanette et al., 2020b; Wagenmaker et al., 2021a) can thus be used to solve PAC RL. However, computationally efficient procedures (Jin et al., 2020b; Wagenmaker et al., 2021a) at best achieve a sample complexity of $\mathcal{O}(d^3 \cdot \text{poly}(H)/\epsilon^2)$. While (Zanette et al., 2020b; Jin et al., 2021) achieve complexities of $\mathcal{O}(d^2 \cdot \text{poly}(H)/\epsilon^2)$, their approaches are computationally inefficient. Note that it is not straightforward to apply these algorithms to the reward-free setting—they rely on access to a reward function during exploration. To our knowledge no lower bounds exist for PAC RL in linear MDPs. As such, it remains an open question what the optimal d dependence is, and if it can be obtained by a computationally efficient algorithm.

3. Preliminaries

Notation. All logarithms log are base-e unless otherwise noted. We let $[m] = \{1, 2, \dots, m\}$, $\mathcal{B}^d(R) := \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\| \le R\}$, the ball of radius R in $\mathbb{R}^d$, and specialize $\mathcal{B}^d := \mathcal{B}^d(1)$ to the unit ball. $\mathcal{S}^{d-1}$ denotes the unit sphere in $\mathbb{R}^d$. $\mathcal{O}(\cdot)$ hides absolute constants, and $\widetilde{\mathcal{O}}(\cdot)$ hides absolute constants and logarithmic terms. Throughout, bold characters refer to vectors and matrices and standard characters refer to scalars.

3.1. Markov Decision Processes

We study finite-horizon, episodic Markov Decision Processes with time-varying transition kernel. We denote an MDP by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, H, \{P_h\}_{h=1}^H, \{r_h\}_{h=1}^H)$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ the set of actions, H the horizon, $\{P_h\}_{h=1}^H$ the transition kernel, $P_h : \mathcal{S} \times \mathcal{A} \to \triangle(\mathcal{S})$, and $\{r_h\}_{h=1}^H$ the reward, $r_h : \mathcal{S} \times \mathcal{A} \to [0, 1]$, which we assume is deterministic[1]. We do not include an initial state distribution but instead assume the MDP always starts in state s_1 and encode the initial state distribution in the first transition, which is without loss of generality. We assume that $\{P_h\}_{h=1}^H$ is initially unknown.

A policy, $\pi = \{\pi_h\}_{h=1}^H$, $\pi_h : \mathcal{S} \to \triangle(\mathcal{A})$, is a mapping from states to actions. Given a policy π, an episode starts in state s_1, the agent takes action $a_1 \sim \pi_1(s_1)$, the MDP transitions to state $s_2 \sim P_1(\cdot|s_1, a_1)$, and the agent receives (and observes) reward $r_1(s_1, a_1)$. This process repeats for H steps: at step h, if the agent is in state s_h, they take action $a_h \sim \pi_h(s_h)$, transition to state $s_{h+1} \sim P_h(\cdot|s_h, a_h)$, and receive reward $r_h(s_h, a_h)$. After H steps the episode terminates and restarts at s_1. In the special case when the policy is deterministic—π_h is supported on only one action for each s and h—we will denote this action as $\pi_h(s)$.

We let $\mathbb{E}_h[V](s, a) = \mathbb{E}_{s' \sim P_h(\cdot|s,a)}[V(s')]$, so $\mathbb{E}_h[V](s, a)$ denotes the expected next-state value of V given from state s after playing action a at time h. For a fixed policy π, we let $\mathbb{E}_\pi[\cdot]$ denote the expectation over the trajectory $(s_1, a_1, \dots, s_H, a_H)$ induced by π on the MDP.

Value Functions. For a policy π, the Q-value function, $Q_h^\pi : \mathcal{S} \times \mathcal{A} \to [0, H]$ is defined as the expected reward one would obtain if action a is taken in state s at step h and then π is followed for all subsequent steps. Precisely,

$$Q_h^\pi(s, a) = \mathbb{E}_\pi \left[\sum_{h'=h}^{H} r_{h'}(s_{h'}, a_{h'}) | s_h = s, a_h = a \right].$$

The value function, $V_h^\pi : \mathcal{S} \to [0, H]$, is defined as the expected reward one would obtain if policy π is played from state s at step h onwards. In terms of the Q-value function, $V_h^\pi(s) = \mathbb{E}_{a \sim \pi_h(s)}[Q_h^\pi(s, a)]$. The Q-value function and value function can be related by the *Bellman equation*:

$$Q_h^\pi(s, a) = r_h(s, a) + \mathbb{E}_h[V_{h+1}^\pi](s, a).$$

For simplicity and to ensure this relationship holds for all $h \in [H]$, we define $V_{H+1}^\pi(s) = Q_{H+1}^\pi(s, a) := 0$ for all s, a, and π. We denote the *value* of a policy by $V_0^\pi := \mathbb{E}_\pi[V_1^\pi(s_1)]$, the expected reward π obtains in an episode. The optimal policy, $\pi^\star$, is the policy which achieves the largest expected reward: $V_0^{\pi^\star} = \sup_\pi V_0^\pi$ (note that $\pi^\star$ need not be unique). We will denote the value function and Q-value function associated with $\pi^\star$ as $V_h^\star(s)$ and $Q_h^\star(s, a)$, respectively. Note that $V^\star$ and $Q^\star$ satisfy $V_h^\star(s) = \sup_\pi V_h^\pi(s)$ and $Q_h^\star(s, a) = \sup_\pi Q_h^\pi(s, a)$.

The value function depends on both the MDP and the reward function. In cases where we want to make this dependence explicit, we denote $V_0^\pi(r)$ the value of policy π *with respect to reward function* r, and similarly define $V_0^\star(r)$ as the value of the optimal policy with respect to r.

Reward-Aware RL (PAC Policy Identification). In the reward-aware RL setting, given a reward function r, the goal is to identify a policy $\widehat{\pi}$ such that, with probability at least $1 - \delta$,

$$V_0^\star(r) - V_0^{\widehat{\pi}}(r) \le \epsilon \tag{3.1}$$

using as few episodes as possible. We call a policy $\widehat{\pi}$ satisfying (3.1) ϵ-*optimal*. Critically, in reward-aware RL the learner has access to rewards while exploring. Note that this setting is also commonly referred to as the *PAC RL* (Probably Approximately Correct) setting in the literature.

Reward Free RL. The reward-free RL setting stands in contrast to the reward-aware RL setting in that the learner does not observe the rewards (or have any access to the reward function) while exploring. The reward-free RL setting proceeds in two phases:

1. Given (ϵ, δ), without observing any rewards, the learner explores an MDP for K episodes, where K is of the learner's choosing, collecting trajectories $\{(s_{1,k}, a_{1,k}, s_{2,k}, a_{2,k}, \dots, s_{H,k}, a_{H,k})\}_{k=1}^K$.
2. The learner outputs a map $\widehat{\pi}(\cdot)$ which takes as input a reward function and outputs a policy such that $V_0^\star(r) - V_0^{\widehat{\pi}(r)}(r) \le \epsilon$ for all valid reward functions r simultaneously.

As in reward-aware RL, the goal is to obtain a procedure able to provide the above guarantee with probability at least $1 - \delta$, and do so using as few episodes as possible.

[1] The assumption that the rewards are deterministic is for expositional convenience only—all our results could easily be modified to handle random rewards.

3.2. Reinforcement Learning with Linear Function Approximation

The RL theory literature has often considered the *tabular* setting, where $|\mathcal{S}| < \infty$, $|\mathcal{A}| < \infty$. While convenient to work with theoretically, this setting is quite limited in practice, and is not a realistic model of real world settings where the state spaces may be infinite, and "nearby" states may behave similarly. Recently, the RL theory community has begun relaxing the tabular assumption and studying large state-space settings. In this work, we consider the linear MDP setting of (Jin et al., 2020b). Linear MDPs are defined as follows.

Definition 3.1 (Linear MDPs)**.** We say that an MDP is a *d-dimensional linear MDP*, if there exists some (known) feature map $\boldsymbol{\phi}(s,a) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$, H (unknown) signed vector-valued measures $\boldsymbol{\mu}_h \in \mathbb{R}^d$ over $\mathcal{S}$, and H (unknown) reward vectors $\boldsymbol{\theta}_h \in \mathbb{R}^d$, such that:

$$P_h(\cdot|s,a) = \langle \boldsymbol{\phi}(s,a), \boldsymbol{\mu}_h(\cdot)\rangle, \quad r_h(s,a) = \langle \boldsymbol{\phi}(s,a), \boldsymbol{\theta}_h\rangle.$$

We will assume $\|\boldsymbol{\phi}(s,a)\|_2 \le 1$ for all s,a; and for all h, $\||\boldsymbol{\mu}_h|(\mathcal{S})\|_2 = \|\int_{s\in\mathcal{S}} |\mathrm{d}\boldsymbol{\mu}_h(s)|\|_2 \le \sqrt{d}$ and $\|\boldsymbol{\theta}_h\|_2 \le \sqrt{d}$.

Jin et al. (2020b) show that the linear MDP setting encompasses tabular MDPs (with $d = |\mathcal{S}||\mathcal{A}|$). Critically, however, it also encompasses MDPs with infinite state-spaces, for instance, MDPs where the feature space is the d-dimensional simplex. In addition, the linear MDP assumptions allows for "nearby" states, states with similar feature vectors, to behave in similar ways, allowing us to generalize across states without visiting every state.

Finally, we introduce the concept of set visitations, which will play an important role in our analysis.

Definition 3.2 (Set Visitation)**.** For any $\mathcal{X} \subseteq \mathbb{R}^d$ and policy π, we let

$$\omega_h^\pi(\mathcal{X}) := \mathbb{P}_\pi[\boldsymbol{\phi}(s_h,a_h) \in \mathcal{X}]$$

denote the probability of visiting $\mathcal{X}$ under π at step h. Furthermore, we will say that $\mathcal{X}$ is *c-unreachable* if $\sup_\pi w_h^\pi(\mathcal{X}) \le c$.

4. Main Results

We first present our main upper bound on reward-free RL, and then provide a lower bound on reward-aware RL—giving a lower bound on reward-free RL as an immediate corollary.

4.1. Upper Bounds on Reward Free RL in Linear MDPs

We present our main algorithm, RFLIN-EXPLORE, as Algorithm 1.

Algorithm 1 Reward Free Linear RL: Exploration (RFLIN-EXPLORE)

input: tolerance ϵ, confidence δ
$K_{\max} \leftarrow \mathrm{poly}(1/\epsilon, d, H, \log 1/\delta)$
$\beta \leftarrow cH\sqrt{d\log(1+dHK_{\max}) + \log H/\delta}$
$\iota_\epsilon \leftarrow \lceil \log_2(4\beta H/\epsilon)\rceil$
$\boldsymbol{\gamma}^2 \in \mathbb{R}^{\iota_\epsilon}$ with $\boldsymbol{\gamma}_i^2 \leftarrow 2^{2i}\cdot \frac{\epsilon^2}{64H^2\iota_\epsilon^2\beta^2}$ for $i \in [\iota_\epsilon]$.
for $h = 1,\dots,H$ **do**
  $\{(\mathcal{X}_{hi}, \mathcal{D}_{hi}, \boldsymbol{\Lambda}_{hi})\}_{i=1}^{\iota_\epsilon} \leftarrow$ COVERTRAJ$(h, \frac{\delta}{H}, \boldsymbol{\gamma}^2, \iota_\epsilon)$
$\mathfrak{D} \leftarrow \{\{(\mathcal{X}_{hi}, \mathcal{D}_{hi}, \boldsymbol{\Lambda}_{hi})\}_{i=1}^{\iota_\epsilon}\}_{h=1}^{H}$
return RFLIN-PLAN$(\cdot\,; \epsilon, \delta, \mathfrak{D})$

RFLIN-EXPLORE relies on a core subroutine, COVERTRAJ, to collect data. We provide a detailed description of COVERTRAJ in Section 5 but give a brief description below.

COVERTRAJ Subroutine. COVERTRAJ generates a set of *covering trajectories*. For a given h, a call to COVERTRAJ partitions the feature space into sets $\mathcal{X}_{hi}$ such that

$$\sup_\pi w_h^\pi(\mathcal{X}_{hi}) \le 2^{-i+1}$$

and collects data $\mathcal{D}_{hi} = \{(s_{h,\tau}^i, a_{h,\tau}^i, s_{h+1,\tau}^i)\}_{\tau=1}^{K_i}$ and covariates $\boldsymbol{\Lambda}_{hi} = I + \sum_{\tau=1}^{K_i} \boldsymbol{\phi}_{h,\tau}^i (\boldsymbol{\phi}_{h,\tau}^i)^\top$ for $\boldsymbol{\phi}_{h,\tau}^i := \boldsymbol{\phi}(s_{h,\tau}^i, a_{h,\tau}^i)$ such that

$$\boldsymbol{\phi}^\top \boldsymbol{\Lambda}_{hi}^{-1} \boldsymbol{\phi} \le \boldsymbol{\gamma}_i^2, \quad \forall \boldsymbol{\phi} \in \mathcal{X}_{hi}.$$

In words, each $\mathcal{X}_{hi}$ corresponds to a set of feature vectors that are 2^{-i+1}-unreachable, and for which we can guarantee a desired amount of data pointing in similar directions has been collected. If a particular direction is difficult to reach, then it is unlikely any policy will encounter it and, as such, to find a near-optimal policy, it suffices to use a relatively large learning tolerance in that direction. Motivated by this, we set $\boldsymbol{\gamma}_i = \mathcal{O}(2^i\epsilon)$—as the sets become more difficult to reach, we require that less data is collected from them.

RFLIN-PLAN Mapping. RFLIN-EXPLORE returns RFLIN-PLAN$(\cdot\,; \epsilon, \delta, \mathfrak{D})$ which defines a mapping from reward functions to policies. This mapping is parameterized by the data returned by COVERTRAJ and, given a reward function as input, outputs a policy it believes is near-optimal for this reward function. RFLIN-PLAN itself runs a simple least squares value iteration procedure to compute the policy. We define RFLIN-PLAN in Algorithm 2. We then have the following result.

Theorem 1. *Consider running* RFLIN-EXPLORE *with tolerance $\epsilon > 0$ and confidence $\delta > 0$, and let* RFLIN-PLAN$(\cdot)$ *denote its output. Then with probability at least $1-\delta$, for an arbitrary number of reward functions r satisfying Definition 3.1,* RFLIN-PLAN(r) *returns a policy*

Algorithm 2 Reward Free Linear RL: Planning (RFLIN-PLAN)

input: reward functions r
parameters: tolerance ϵ, confidence δ, data $\{\{(\mathcal{X}_{hi}, \mathcal{D}_{hi}, \mathbf{\Lambda}_{hi})\}_{i=1}^{\iota_\epsilon}\}_{h=1}^{H}$
$K_{\max} \leftarrow \text{poly}(1/\epsilon, d, H, \log 1/\delta)$
$\beta \leftarrow cH\sqrt{d\log(1 + dHK_{\max}) + \log H/\delta}$
for $h = H, H-1, \ldots, 1$ **do**
  $\widehat{\boldsymbol{w}}_h \leftarrow \arg\min_{\boldsymbol{w}} \sum_{i=1}^{\iota_\epsilon}\sum_{\tau=1}^{K_i}\left(\boldsymbol{w}^\top \boldsymbol{\phi}_{h,\tau}^i - r_h(s_{h,\tau}^i, a_{h,\tau}^i) - V_{h+1}(s_{h+1,\tau}^i)\right)^2 + \|\boldsymbol{w}\|_2^2$
  $\mathbf{\Lambda}_h = I + \sum_{i=1}^{\iota_\epsilon}\sum_{\tau=1}^{K_i}\boldsymbol{\phi}_{h,\tau}^i(\boldsymbol{\phi}_{h,\tau}^i)^\top$
  $Q_h(\cdot,\cdot) \leftarrow \min\{\langle\boldsymbol{\phi}(\cdot,\cdot), \widehat{\boldsymbol{w}}_h\rangle + \beta\|\boldsymbol{\phi}(\cdot,\cdot)\|_{\mathbf{\Lambda}_h^{-1}}, H\}$
  $V_h(\cdot) \leftarrow \max_a Q_h(\cdot, a)$
  $\widehat{\pi}_h(\cdot) \leftarrow \arg\max_a Q_h(\cdot, a)$
return $\{\widehat{\pi}_h\}_{h=1}^{H}$

$\widehat{\pi}$ which is ϵ-optimal with respect to r. Furthermore, this procedure collects at most

$$\widetilde{\mathcal{O}}\left(\frac{dH^5(d+\log 1/\delta)}{\epsilon^2} + \frac{d^{9/2}H^6\log^4(1/\delta)}{\epsilon}\right)$$

episodes, where $\widetilde{\mathcal{O}}(\cdot)$ hides absolute constants and terms $\text{poly}\log(d, H, 1/\epsilon)$ *and* $\text{poly}\log\log(1/\delta)$.

For ϵ sufficiently small, exploring via RFLIN-EXPLORE and planning via RFLIN-PLAN learns an ϵ-optimal policy for an arbitrarily large number of reward functions using only $\widetilde{\mathcal{O}}(\frac{d^2H^5}{\epsilon^2})$ episodes. This improves on both existing reward-free algorithms for linear MDPs by a factor of d (Zanette et al., 2020c; Wang et al., 2020).

Remark 4.1 (Computational Efficiency)**.** In MDPs with a finite number of actions, both RFLIN-EXPLORE and RFLIN-PLAN are computationally efficient, with computational cost scaling polynomially in $d, H, 1/\epsilon, \log 1/\delta$, and $|\mathcal{A}|$. The primary computational cost of RFLIN-EXPLORE is due to COVERTRAJ, while the computational cost of RFLIN-PLAN is due primarily to solving a least-squares problem. The computational cost of COVERTRAJ is described in more detail in Section 5, but is dominated by calls to a computationally efficient regret minimization algorithm. In the case when an infinite number of actions is available, the $|\mathcal{A}|$ dependence can be replaced with $2^{\mathcal{O}(d)}$. As several existing works have noted, this dependence seems unavoidable (Jin et al., 2020b; Wagenmaker et al., 2021a).

Remark 4.2 (Reward-Aware RL)**.** When only a single reward function is given, we recover the standard reward-aware RL setting. Hence, RFLIN-EXPLORE and RFLIN-PLAN provide a computationally efficient reward-aware RL algorithm that learns an ϵ-optimal policy after collecting $\widetilde{\mathcal{O}}(\frac{dH^5(d+\log 1/\delta)}{\epsilon^2})$ episodes. Our more general *reward-free* result is sharper than the less-general reward-aware results derived from an online-to-batch conversion of prior computationally efficient low-regret algorithms (Jin et al., 2020b; Wagenmaker et al., 2021a) by a factor of d. Our result also improves on the computationally-inefficient $\mathcal{O}(\frac{d^2H^4}{\epsilon^2})$ reward-aware bound due to (Zanette et al., 2020b) in that (a) it is computationally efficient, and (b) it extends to the reward-free setting.

Remark 4.3 (Nonlinear Rewards)**.** Our procedure is able to handle nonlinear reward functions with little modification. The primary difference in the setting of nonlinear rewards is that we must have query access to the reward function for *all* s, a, h. In contrast, if the reward functions are linear, our procedure only needs to know the reward function values for the trajectories observed during exploration.

4.2. Lower Bounds on Reward-Aware RL in Linear MDPs

To our knowledge, Theorem 1 is the first result to show that computationally efficient d^2 complexity is possible in linear MDPs for any of the reward-free, reward-aware, or regret minimization settings. We next show a somewhat surprising result: reward-free RL is no harder than reward-aware RL in linear MDPs, up to horizon factors. To this end, we show a lower bound on reward-aware RL in linear MDPs. As a warm-up, we first provide a lower bound for the simpler linear bandit setting.

Definition 4.1 (Linear Bandits and ϵ-Optimal Arms)**.** Consider the linear bandit setting parameterized by some $\boldsymbol{\theta} \in \mathbb{R}^d$ and $\Phi \subseteq \mathbb{R}^d$, where at every step k the learner chooses $\boldsymbol{\phi}_k \in \Phi$ and observes

$$y_k \sim \text{Bernoulli}(\langle\boldsymbol{\theta}, \boldsymbol{\phi}_k\rangle + 1/2).$$

We say an arm $\boldsymbol{\phi} \in \Phi$ is ϵ-optimal if

$$\langle\boldsymbol{\theta}, \boldsymbol{\phi}\rangle + 1/2 \geq \sup_{\boldsymbol{\phi}' \in \Phi}\langle\boldsymbol{\theta}, \boldsymbol{\phi}'\rangle + 1/2 - \epsilon$$

and a policy $\pi \in \triangle_\Phi$ is ϵ-optimal if

$$\mathbb{E}_{\boldsymbol{\phi}\sim\pi}[\langle\boldsymbol{\theta}, \boldsymbol{\phi}\rangle] + 1/2 \geq \sup_{\boldsymbol{\phi}' \in \Phi}\langle\boldsymbol{\theta}, \boldsymbol{\phi}'\rangle + 1/2 - \epsilon.$$

Theorem 2. *Fix $\epsilon > 0$, $d > 1$, and $K \geq d^2$. Let $\boldsymbol{\theta} \in \Theta = \{-\sqrt{d/700K}, \sqrt{d/700K}\}^d$ and $\Phi = \mathcal{S}^{d-1}$. Consider running a (possibly adaptive) algorithm in the linear bandit setting of Definition 4.1 which stops at (a possibly random stopping time) τ and outputs a guess at an ϵ-optimal policy, $\widehat{\pi} \in \triangle_\Phi$. Let $\mathcal{E}$ be the event*

$$\mathcal{E} := \{\tau \leq K \text{ and } \widehat{\pi} \text{ is } \epsilon\text{-optimal}\}.$$

Then unless $K \geq c \cdot \frac{d^2}{\epsilon^2}$ for a universal $c > 0$, there exists $\boldsymbol{\theta} \in \Theta$ for which $\mathbb{P}_{\boldsymbol{\theta}}[\mathcal{E}^c] \geq 1/10$; i.e., with constant probability either $\widehat{\pi}$ is not ϵ-optimal or more than K samples are collected.

Setting $K = \frac{c}{2} \cdot \frac{d^2}{\epsilon^2}$, Theorem 2 implies that with constant probability, any algorithm will gather either more than $\frac{c}{2} \cdot \frac{d^2}{\epsilon^2}$ samples, or will output a policy which is not ϵ-optimal. While this shows that $\Omega(d^2/\epsilon^2)$ samples is necessary to find an ϵ-optimal policy in linear bandits, there is a slight discrepancy between the model class in Definition 4.1 and the linear MDP setting, Definition 3.1. In the former, the rewards are assumed to be random, while in the latter, the rewards are deterministic. Nevertheless, we show in Appendix D.1 that it is possible to encode the linear bandit structure of Definition 4.1 in a linear MDP with deterministic, known rewards. This yields the following corollary.

Corollary 1. *Fix $\epsilon > 0$, $d > 1$, $H > 1$, and $K \geq d^2$. Consider running a (possibly adaptive) algorithm for K episodes in a $(d+1)$-dimensional linear MDP with horizon H, which stops at (a possibly random stopping time) τ and outputs a guess at an ϵ-optimal policy, $\widehat{\pi}$. Let $\mathcal{E}$ be the event*

$$\mathcal{E} := \{\tau \leq K \text{ and } \widehat{\pi} \text{ is } \epsilon\text{-optimal}\}.$$

Then there is a universal constant $c > 0$ such that unless

$$K \geq c \cdot \frac{d^2 H^2}{\epsilon^2},$$

there exists a linear MDP $\mathcal{M}$ for which $\mathbb{P}_{\mathcal{M}}[\mathcal{E}^c] \geq 1/10$.

As Corollary 1 shows, a $\Omega(d^2H^2/\epsilon^2)$ dependence is necessary for learning an ϵ-optimal policy in linear MDPs. Note that this lower bound holds for learning a *single* policy given access to the reward function during exploration (indeed, it holds in the case when the learner has oracle access to the rewards)—the reward-aware RL setting.

In contrast, Theorem 1 shows that we can learn ϵ-optimal policies for *all valid reward functions simultaneously*, without having access to the reward functions during exploration, using only $\widetilde{\mathcal{O}}(d^2H^5/\epsilon^2)$ episodes. In other words, up to H factors, ***reward-free RL is no harder than reward-aware RL in linear MDPs***. This is in contrast to the tabular setting, where it is well known that the optimal rate for reward-aware RL is $\Theta(|\mathcal{S}||\mathcal{A}|/\epsilon^2)$ while the optimal rate for reward-free RL is $\Theta(|\mathcal{S}|^2|\mathcal{A}|/\epsilon^2)$.

Remark 4.4 (H Dependence)**.** Our claim that "reward-free RL is no harder than reward-aware RL in linear MDPs" only holds up to H factors—as Theorem 1 and Corollary 1 show, there is still a discrepancy in the H dependence for our reward-free upper bound and reward-aware lower bound. In the tabular setting, the difference in hardness between the reward-free and reward-aware setting lies in the "dimensionality" factors (that is, $|\mathcal{S}|$)—the optimal H dependence for each is identical (compare the reward-free rate of (Zhang et al., 2020a) with the reward-aware rate of (Zhang et al., 2020b)). Thus, our result shows that the difference in hardness between the reward-free and reward-aware problems which is present in the tabular setting is not present in the linear setting, motivating our claim.

We do not believe the H dependence of Theorem 1 is optimal, and also conjecture that the H dependence in the lower bound of Corollary 1 can be increased, to obtain matching H dependence in our upper and lower bounds. As the goal of this paper is optimizing d factors and not H factors, we did not focus on improving the H dependence, and leave this for future work.

5. Efficient Exploration in Linear MDPs

Our main algorithmic technique is an exploration procedure able to efficiently generate "covering trajectories" in linear MDPs, which we believe may be of independent interest. Before presenting our MDP covering algorithm, COVERTRAJ, we describe its key subroutine, EGS.

Algorithm 3 Explore Goal Set (EGS)

1: **input:** goal set $\mathcal{X} \subseteq \mathbb{R}^d$, step h, number of episodes K, collection tolerance γ^2, regret minimization algorithm REGMIN
2: Run REGMIN for K episodes, using reward $r_h^k(s,a) := r(\boldsymbol{\phi}(s,a); \boldsymbol{\Lambda}_{h,k-1})$ at episode k, for r defined as in (5.1), $\boldsymbol{\Lambda}_{h,k-1} = I + \sum_{\tau=1}^{k-1} \boldsymbol{\phi}_{h,\tau}\boldsymbol{\phi}_{h,\tau}^\top$, and $r_{h'}^k(s,a) = 0$ for $h' \neq h$
3: Collect transitions observed at step h

$$\mathcal{D} \leftarrow \{(s_{h,\tau}, a_{h,\tau}, s_{h+1,\tau})\}_{\tau=1}^K$$

4: **return** $\{\boldsymbol{\phi} \in \mathcal{X} \ : \ \boldsymbol{\phi}^\top \boldsymbol{\Lambda}_{h,K}^{-1} \boldsymbol{\phi} \leq \gamma^2\}$, $\mathcal{D}$, $\boldsymbol{\Lambda}_{h,K}$

Explore Goal Set (EGS) Subroutine. EGS takes as input a target set $\mathcal{X}$ to be explored; we fix $\mathcal{X}$ in what follows. It then creates an "exploration reward function"—which places a high reward on regions of $\mathcal{X}$ for which we have large uncertainty—and then runs a regret minimization algorithm on this reward function to direct exploration to these regions. More specifically, we define the reward function

$$r(\boldsymbol{\phi}; \boldsymbol{\Lambda}) \leftarrow \begin{cases} 1 & \|\boldsymbol{\phi}\|_{\boldsymbol{\Lambda}^{-1}}^2 > \gamma^2, \boldsymbol{\phi} \in \mathcal{X} \\ \frac{1}{\gamma^2}\|\boldsymbol{\phi}\|_{\boldsymbol{\Lambda}^{-1}}^2 & \|\boldsymbol{\phi}\|_{\boldsymbol{\Lambda}^{-1}}^2 \leq \gamma^2, \boldsymbol{\phi} \in \mathcal{X} \\ 0 & \boldsymbol{\phi} \notin \mathcal{X} \end{cases}. \quad (5.1)$$

At the kth episode, we instantiate our reward as $r_h^k(s,a) := r(\boldsymbol{\phi}(s,a); \boldsymbol{\Lambda}_{h,k-1})$. This captures the fact that we have collected covariates $\boldsymbol{\Lambda}_{h,k-1} = I + \sum_{\tau=1}^{k-1} \boldsymbol{\phi}_{h,\tau}\boldsymbol{\phi}_{h,\tau}^\top$ over the first $k-1$ episodes, so our uncertainty in direction $\boldsymbol{\phi}$ scales as $\|\boldsymbol{\phi}\|_{\boldsymbol{\Lambda}_{h,k-1}^{-1}}$. By choosing our reward function to increase as $\|\boldsymbol{\phi}\|_{\boldsymbol{\Lambda}_{h,k-1}^{-1}}$ increases (for $\boldsymbol{\phi} \in \mathcal{X}$), we incentive exploring directions in $\boldsymbol{\phi} \in \mathcal{X}$ with large uncertainty. EGS also takes as input a scalar tolerance γ^2 and, after running

for K episodes, returns the set of $\phi \in \mathcal{X}$ which have been explored up to tolerance γ^2.

In order to efficiently collect data, we require a regret-minimization algorithm which achieves low regret with respect to a *time-varying* reward function, r^k. We define regret with respect to such a reward function as

$$\mathcal{R}_K := \sum_{k=1}^K [V_0^\star(r^k) - V_0^{\pi_k}(r^k)].$$

To achieve the optimal scaling in d, REGMIN must attain *first order* regret in the following sense.

Definition 5.1 (First-Order Regret Minimization Algorithm)**.** Consider a time-varying reward function r^k that is $\mathcal{F}_{k-1}$-measurable, satisfies $r_h^k(s,a) \in [0,1]$, and is non-increasing in k (that is, $r_h^k(s,a) \le r_h^{k-1}(s,a)$ for all s,a,h,k). Then we call a regret minimization algorithm REGMIN a *first-order regret minimization algorithm* if it achieves regret bounded as, with probability at least $1-\delta$,

$$\mathcal{R}_K \le \sqrt{\mathcal{C}_1 V_0^\star(r^1) K \cdot \log^{p_1}(HK/\delta)} + \mathcal{C}_2 \log^{p_2}(HK/\delta)$$

for constants $\mathcal{C}_1, \mathcal{C}_2, p_1, p_2$ which do not depend on K.

In general, we can think of $\mathcal{C}_1$ and $\mathcal{C}_2$ as $\mathrm{poly}(d,H)$ and p_1 and p_2 as absolute constants. We are now ready to present our main exploration algorithm, COVERTRAJ, in Algorithm 4.

Algorithm 4 Collect Covering Trajectories (COVERTRAJ)

input: step h, confidence δ, number of epochs m, tolerance vector $\boldsymbol{\gamma}^2 \in [0,1]^m$, regret minimization algorithm REGMIN (default: FORCE, see Section 5.1)
$\mathcal{X} \leftarrow \mathcal{B}^d$
for $i = 1, 2, \dots, m$ **do**
  Set K_i as in (5.2)
  $\mathcal{X}_i, \mathcal{D}_i, \boldsymbol{\Lambda}_i \leftarrow \textsc{Egs}(\mathcal{X}, h, K_i, \boldsymbol{\gamma}_i^2, \textsc{RegMin})$
  $\mathcal{X} \leftarrow \mathcal{X} \backslash \mathcal{X}_i$
return $\{(\mathcal{X}_i, \mathcal{D}_i, \boldsymbol{\Lambda}_i)\}_{i=1}^m$

COVERTRAJ Algorithm. COVERTRAJ partitions the feature space by repeatedly calling EGS while exponentially increasing the number of episodes EGS is run for. As the number of episodes EGS is run for increases, EGS is able to reach harder and harder to reach feature directions, while ensuring easier to reach directions have been explored up to their desired tolerance $\boldsymbol{\gamma}_i^2$. Specifically, at each epoch i, EGS runs for

$$K_i \leftarrow \widetilde{\mathcal{O}}\Bigg(2^i \cdot \max\Bigg\{\frac{d}{\boldsymbol{\gamma}_i^2} \log \tfrac{2^i}{\boldsymbol{\gamma}_i^2}, \mathcal{C}_1 (mp_1)^{p_1} \log^{p_1} \tfrac{1}{\delta}, \mathcal{C}_2 (mp_2)^{p_2} \log^{p_2} \tfrac{1}{\delta}\Bigg\}\Bigg) \tag{5.2}$$

episodes. COVERTRAJ enjoys the following guarantee.

Theorem 3. *Fix $h \in [H]$, $\delta > 0$, $m \ge 1$, and set $\boldsymbol{\gamma}^2 \in [0,1]^m$ to desired tolerances. Consider running* COVERTRAJ *with a regret minimization algorithm* REGMIN *satisfying Definition 5.1, and let $\{(\mathcal{X}_i, \mathcal{D}_i, \boldsymbol{\Lambda}_i)\}_{i=1}^m$ denote the arguments returned. Then, with probability at least $1-\delta$, for each $i \in [m]$ simultaneously:*

$$\sup_\pi w_h^\pi(\mathcal{X}_i) \le 2^{-i+1} \quad \text{and} \quad \boldsymbol{\phi}^\top \boldsymbol{\Lambda}_i^{-1} \boldsymbol{\phi} \le \boldsymbol{\gamma}_i^2, \forall \boldsymbol{\phi} \in \mathcal{X}_i$$

and, on the same event, it also holds that

$$\sup_\pi w_h^\pi(\mathcal{B}^d \backslash \cup_{i=1}^m \mathcal{X}_i) \le 2^{-m}.$$

Furthermore, COVERTRAJ *terminates after at most $\sum_{i=1}^m K_i$ episodes, for K_i as in (5.2).*

Theorem 3 shows that COVERTRAJ partitions the feature space into sets $\mathcal{X}_i$ such that $\mathcal{X}_i$ is 2^{-i+1}-unreachable, and where we have learned every $\boldsymbol{\phi} \in \mathcal{X}_i$ up to tolerance $\boldsymbol{\gamma}_i^2$: $\|\boldsymbol{\phi}\|_{\boldsymbol{\Lambda}_i^{-1}}^2 \le \boldsymbol{\gamma}_i^2$. Furthermore, it takes roughly $2^i \cdot \frac{d}{\boldsymbol{\gamma}_i^2}$ episodes of exploration to accomplish this, which is the intuitively correct rate, as the following example illustrates.

Example 5.1 (Tabular MDPs)**.** Consider a tabular MDP with $H = 2$, state space $\mathcal{S}$ with $|\mathcal{S}| < \infty$, and A actions (we think of $A \ll |\mathcal{S}|$ as an absolute constant). Assume we always start in state s_0 and can break the state space into sets $\mathcal{S}_1, \dots, \mathcal{S}_A$ such that $|\mathcal{S}_1| = \mathcal{O}(|\mathcal{S}|/A)$ and where, if we take action a_i, we will end up in $\mathcal{S}_i$ with probability 2^{-i} (with equal probability of being in any particular state within $\mathcal{S}_i$), and s_0 with probability $1 - 2^{-i}$. Representing this as a linear MDP, we have $d = A|\mathcal{S}|$ and $\boldsymbol{\phi}(s,a) = \boldsymbol{e}_{sa}$, a standard basis vector. As such, $\boldsymbol{\Lambda}_K$, the covariates collected over K episodes, is diagonal with $[\boldsymbol{\Lambda}_K]_{sa} = N(s,a)$ the number of visits to s,a.

Now assume we want to ensure that $\boldsymbol{\phi}(s,a)^\top \boldsymbol{\Lambda}_K^{-1} \boldsymbol{\phi}(s,a) \le \boldsymbol{\gamma}_i^2$ for some $\boldsymbol{\gamma}_i^2$, each $s \in \mathcal{S}_i$, and all $a \in [A]$. For any given $s \in \mathcal{S}_i$, if we take action a_i in state s_0, we will arrive in state s with probability $2^{-i}/|\mathcal{S}_i|$. Thus, in expectation, to collect N samples from s, we must run for at least

$$\tfrac{N}{2^{-i}/|\mathcal{S}_i|} = 2^i |\mathcal{S}_i| N$$

episodes. It follows that to collect N samples from each $s \in \mathcal{S}_i$ and all $a \in [A]$, it will take at least $2^i |\mathcal{S}_i| AN$ episodes. Furthermore, by the above observation that $[\boldsymbol{\Lambda}_K]_{sa} = N(s,a)$, achieving $\boldsymbol{\phi}(s,a)^\top \boldsymbol{\Lambda}_K^{-1} \boldsymbol{\phi}(s,a) \le \boldsymbol{\gamma}_i^2$ is equivalent to setting $N = 1/\boldsymbol{\gamma}_i^2$. Since each $\mathcal{S}_i$ can only be reached independently of the other $\mathcal{S}_j$, $j \ne i$, this implies that to meet our objective we must run for at least

$$\sum_{i=1}^A 2^i \cdot \tfrac{|\mathcal{S}_i| A}{\boldsymbol{\gamma}_i^2} = \mathcal{O}\Big(\sum_{i=1}^A 2^i \cdot \tfrac{d}{\boldsymbol{\gamma}_i^2}\Big)$$

episodes. This recovers the complexity COVERTRAJ gets as given in Theorem 3 (for $\boldsymbol{\gamma}_i^2$ sufficiently small).

5.1. Instantiating CoverTraj with Force

A recent work, (Wagenmaker et al., 2021a) provides a first-order regret minimization algorithm for linear MDPs, Force. The following result shows the complexity of CoverTraj when instantiated with Force.

Corollary 2. *When running* CoverTraj *with* RegMin *set to the computationally efficient version of* Force *(Wagenmaker et al., 2021a), all the guarantees of Theorem 3 hold, and the sample complexity can be bounded as*

$$\widetilde{\mathcal{O}}\left(\sum_{i=1}^{m} 2^i \cdot \max\left\{\frac{d}{\gamma_i^2}\log\frac{2^i}{\gamma_i^2}, d^4H^3m^3\log^{7/2}\frac{1}{\delta}\right\}\right)$$

where the $\widetilde{\mathcal{O}}(\cdot)$ hides absolute constants and logarithmic terms. Furthermore, when run with Force, CoverTraj *is computationally efficient with computational cost scaling polynomially in problem parameters.*

Thus, CoverTraj may be instantiated in a computationally efficient way. We make several additional comments on the computational complexity of this approach.

Computational Complexity. Note that the primary computational cost of CoverTraj is incurred in running RegMin, and in evaluating the reward function. As the sets $\mathcal{X}_i$ can be parameterized as ellipsoids, it can be efficiently checked if a given $\phi \in \mathbb{R}^d$ is in $\mathcal{X}_i$, which makes evaluating the reward function efficient. Furthermore, Force only needs access to the reward function at points encountered on each trajectory, so it only need evaluate the reward at polynomially many points. As noted, a computationally efficient version of Force exists (assuming $|\mathcal{A}|$ is not too large), which is what we employ here, making the entire procedure computationally efficient.

5.2. Deriving Theorem 1 from CoverTraj

We briefly sketch out how one may apply CoverTraj to obtain Theorem 1 and defer the full proof to Appendix C. Consider running RFLin-Explore and then calling RFLin-Plan. Let V_h denote the estimates maintained by RFLin-Plan. We first apply the following self-normalized bound.

Lemma 5.1. *With high probability,*

$$\left\|\sum_{i=1}^{\iota_\epsilon}\sum_{\tau=1}^{K_i}\phi^i_{h,\tau}[V_{h+1}(s^i_{h+1,\tau}) - \mathbb{E}_h[V_{h+1}](s^i_{h,\tau}, a^i_{h,\tau})]\right\|_{\mathbf{\Lambda}_h^{-1}} \le cH\sqrt{d\log(1+dHK_{\max}) + \log H/\delta} = \beta.$$

Critically, in contrast to the self-normalized bound used in (Jin et al., 2020b), ours scales as $\widetilde{\mathcal{O}}(H\sqrt{d})$ instead of $\widetilde{\mathcal{O}}(Hd)$. As our exploration procedure explores each $h \in [H]$ separately, V_{h+1} is *uncorrelated* with $\{\{(s^i_{h,\tau}, a^i_{h,\tau}, s^i_{h+1,\tau})\}_{\tau=1}^{K_i}\}_{i=1}^{\iota_\epsilon}$, and we can avoid the union bound over $\mathbf{\Lambda}_{h+1}$ that is necessary in (Jin et al., 2020b), saving us a factor of $\sqrt{d}$.

Given Lemma 5.1, using an argument similar to (Jin et al., 2020b), one can show that

$$V_h(s_h) \le r_h(s_h, \widehat{\pi}_h(s_h)) + \mathbb{E}_h[V_{h+1}](s_h, \widehat{\pi}_h(s_h)) + 2\beta\|\boldsymbol{\phi}(s_h, \widehat{\pi}_h(s_h))\|_{\mathbf{\Lambda}_h^{-1}}$$

and, furthermore, that $V_0 \ge V_0^\star$. Some algebra shows that we can then bound the suboptimality of $\widehat{\pi}$, the policy returned by RFLin-Plan, as

$$V_0^\star - V_0^{\widehat{\pi}} \le V_0 - V_0^{\widehat{\pi}} \le 2\beta\sum_{h=1}^{H}\mathbb{E}_{\widehat{\pi}}[\|\boldsymbol{\phi}(s_h, \widehat{\pi}_h(s_h))\|_{\mathbf{\Lambda}_h^{-1}}].$$

However, it is easy to see that

$$\begin{aligned}
&\sum_{h=1}^{H}\mathbb{E}_{\widehat{\pi}}[\|\boldsymbol{\phi}(s_h, \widehat{\pi}_h(s_h))\|_{\mathbf{\Lambda}_h^{-1}}] \\
&\quad\le \sum_{h=1}^{H}\sup_{\pi}\mathbb{E}_{\pi}[\|\boldsymbol{\phi}(s_h, a_h)\|_{\mathbf{\Lambda}_h^{-1}}] \\
&\quad\le \sum_{h=1}^{H}\sum_{i=1}^{\iota_\epsilon+1}\sup_{\boldsymbol{\phi}\in\mathcal{X}_{hi}}\|\boldsymbol{\phi}\|_{\mathbf{\Lambda}_h^{-1}}\cdot\sup_{\pi}\mathbb{E}_{\pi}[\mathbb{I}\{\boldsymbol{\phi}(s_h, a_h)\in\mathcal{X}_{hi}\}].
\end{aligned}$$

Recall that RFLin-Explore calls CoverTraj for each $h \in [H]$ with tolerances $\gamma_i^2 = \mathcal{O}(\frac{2^{2i}\epsilon^2}{dH^4})$. Applying Theorem 3, we then have that $\sup_\pi \mathbb{E}_\pi[\mathbb{I}\{\boldsymbol{\phi}(s_h, a_h)\in\mathcal{X}_{hi}\}] = \sup_\pi w_h^\pi(\mathcal{X}_{hi}) \le 2^{-i+1}$ and $\sup_{\boldsymbol{\phi}\in\mathcal{X}_{hi}}\|\boldsymbol{\phi}\|_{\mathbf{\Lambda}_h^{-1}} \le \sqrt{\gamma_i^2} = \mathcal{O}(2^i\epsilon/\sqrt{d}H^2)$. Thus, since $\beta = \widetilde{\mathcal{O}}(\sqrt{d}H)$, we can bound the total suboptimality as

$$\begin{aligned}
V_0^\star - V_0^{\widehat{\pi}} &\le 2\beta\sum_{h=1}^{H}\mathbb{E}_{\widehat{\pi}}[\|\boldsymbol{\phi}(s_h, \widehat{\pi}_h(s_h))\|_{\mathbf{\Lambda}_h^{-1}}] \\
&\le \widetilde{\mathcal{O}}(\sqrt{d}H)\cdot\sum_{h=1}^{H}\sum_{i=1}^{\iota_\epsilon+1}\mathcal{O}(2^i\epsilon/\sqrt{d}H^2)\cdot 2^{-i+1} \\
&= \widetilde{\mathcal{O}}(\epsilon)
\end{aligned}$$

so it follows that $\widehat{\pi}$ is ϵ-optimal.

We remark briefly on our choice of γ_i^2. Note that as i increases, the probability of reaching $\mathcal{X}_{hi}$ decreases exponentially in i, $\sup_\pi \mathbb{E}_\pi[\mathbb{I}\{\boldsymbol{\phi}(s_h, a_h) \in \mathcal{X}_{hi}\}] = \sup_\pi w_h^\pi(\mathcal{X}_{hi}) \le 2^{-i+1}$. Thus, if our "uncertainty" over $\mathcal{X}_{hi}$ scales as $\mathcal{O}(2^i)$, the net contribution of $\mathcal{X}_{hi}$ to the suboptimality scales as $\mathcal{O}(1)$. As we can bound our uncertainty over $\mathcal{X}_{hi}$ as $\sup_{\boldsymbol{\phi}\in\mathcal{X}_{hi}}\|\boldsymbol{\phi}\|_{\mathbf{\Lambda}_h^{-1}} \le \sqrt{\gamma_i^2}$, our choice of $\gamma_i^2 = \mathcal{O}(\frac{2^{2i}\epsilon^2}{dH^4})$ results in a contribution to the suboptimality of $\mathcal{O}(\epsilon/\sqrt{d}H^2)$ for $\mathcal{X}_{hi}$—our choice of γ_i^2 cancels the contribution of how easily $\mathcal{X}_{hi}$ can be reached. In other

words, we only need to decrease uncertainty for a given $\mathcal{X}_{hi}$ in proportion to how easily that $\mathcal{X}_{hi}$ can be reached.

Furthermore, our choice of $\boldsymbol{\gamma}_i^2$ allows us to efficiently collect the samples necessary to reduce uncertainty. The leading-order term in the complexity given in Corollary 2 scales as, given our choice of $\boldsymbol{\gamma}_i^2$:

$$\widetilde{\mathcal{O}}\left(\sum_{i=1}^{m} \frac{2^i d}{\boldsymbol{\gamma}_i^2}\right) = \widetilde{\mathcal{O}}\left(\sum_{i=1}^{m} \frac{d^2 H^4}{2^i \cdot \epsilon^2}\right) = \widetilde{\mathcal{O}}\left(\frac{d^2 H^4}{\epsilon^2}\right),$$

Repeated for each h this yields the complexity given in Theorem 1. The key property we exploit here is that CoverTraj collects samples from a given $\mathcal{X}_{hi}$ at a rate inversely proportional to how easily $\mathcal{X}_{hi}$ can be reached—it takes on order $\mathcal{O}(2^i)$ episodes for CoverTraj to collect a sample from $\mathcal{X}_{hi}$, while the probability of reaching $\mathcal{X}_{hi}$ (for any algorithm) is at most on order 2^{-i}. Thus, as our choice of $\boldsymbol{\gamma}_i^2$ only guarantees that we reduce uncertainty for $\mathcal{X}_{hi}$ in proportion with how easily $\mathcal{X}_{hi}$ can be reached, we have that the complexity of collecting the necessary samples is not prohibitively large, and in particular does not scale with the difficulty of reaching $\mathcal{X}_{hi}$.

Remark 5.1 (Necessity of First-Order Regret)**.** Suppose that we instantiate CoverTraj with a regret minimization algorithm which only achieves minimax regret, $\mathcal{R}_K \leq \widetilde{\mathcal{O}}(\sqrt{\mathcal{C}_1 K})$, rather than first-order regret. In order to guarantee $\sup_\pi w_h^\pi(\mathcal{X}_{i+1}) \leq 2^{-i+2}$, we must show that $\mathcal{O}(2^{-i} K_i) \geq \mathcal{R}_{K_i}$, which ensures we have reached the "difficult to reach" states. Using a minimax regret algorithm we therefore need $K_i \geq \widetilde{\mathcal{O}}(2^{2i}\mathcal{C}_1)$, while for a first-order algorithm, assuming $\sup_\pi w_h^\pi(\mathcal{X}_i) \leq 2^{-i+1}$, our choice of reward function gives $V_0^\star(r^1) \leq \mathcal{O}(2^{-i})$, so we need $\mathcal{O}(2^{-i}K_i) \geq \widetilde{\mathcal{O}}(\sqrt{\mathcal{C}_1 2^{-i} K_i}) \iff K_i \geq \widetilde{\mathcal{O}}(2^i \mathcal{C}_1)$. This makes a critical difference when applying CoverTraj to reward-free RL in RFLin-Explore, since in RFLin-Explore we set $m = \iota_\epsilon = \mathcal{O}(\log(\sqrt{d}H^2/\epsilon))$. With this choice of m, a non-first-order algorithm would have complexity $\widetilde{\mathcal{O}}(\sum_{i=1}^m 2^{2i} \cdot \mathcal{C}_1) = \widetilde{\mathcal{O}}(2^{2m} \cdot \mathcal{C}_1) = \widetilde{\mathcal{O}}(\mathcal{C}_1 \frac{d^2 H^4}{\epsilon^2})$. As the best known minimax regret algorithm has $\mathcal{C}_1 = d^2 H^4$, we obtain a (very suboptimal) final complexity of $\widetilde{\mathcal{O}}(\frac{d^4 H^8}{\epsilon^2})$. On the other hand, a first-order algorithm attains $\widetilde{\mathcal{O}}(\sum_{i=1}^m 2^i \cdot \mathcal{C}_1) = \widetilde{\mathcal{O}}(2^m \cdot \mathcal{C}_1) = \widetilde{\mathcal{O}}(\mathcal{C}_1 \frac{dH^2}{\epsilon})$.

5.3. Well-Conditioned Covariates

We conclude with an additional application of CoverTraj to the problem of obtaining well-conditioned covariates. Several existing works (Hao et al., 2021; Agarwal et al., 2021) assume access to a policy π_{exp} able to collect covariates with minimum eigenvalue bounded away from 0, in order to ensure learning in every direction. However, to our knowledge, without access to such an oracle policy, there does not exist an algorithm able to provably collect such "full-rank" data. In the following result, we show that CoverTraj can be used to collect such data, assuming it is possible.

Theorem 4. *Fix $h \in [H]$, $\gamma \in [0,1]$, and suppose $\sup_\pi \lambda_{\min}(\mathbb{E}_\pi[\boldsymbol{\phi}(s_h,a_h)\boldsymbol{\phi}(s_h,a_h)^\top]) \geq \epsilon$. Then there exists an algorithm which collects observations $\mathcal{D}_{\text{exp}} = \{(s_{h,\tau}, a_{h,\tau})\}_{\tau=1}^K$ such that, with probability at least $1-\delta$:*

$$\lambda_{\min}\left(\sum_{(s,a)\in\mathcal{D}_{\text{exp}}} \boldsymbol{\phi}(s,a)\boldsymbol{\phi}(s,a)^\top\right) \geq \frac{\epsilon}{\gamma^2}$$

after running for at most

$$K \leq \widetilde{\mathcal{O}}\left(\frac{1}{\epsilon} \cdot \max\left\{\frac{d}{\gamma^2}, d^4 H^3 \log^3\frac{1}{\delta}\right\}\right)$$

episodes.

6. Conclusion

In this work we have shown that in linear MDPs, reward-free RL is no harder than reward-aware RL. Along the way, we have developed a novel sample collection strategy that allows for efficient traversal of linear MDPs. Several questions remain open for future work. While this work establishes that d^2 is the optimal dimension-dependence for reward-aware (and reward-free) RL, our techniques do not directly provide a regret minimization algorithm. Developing a computationally efficient regret minimization algorithm with regret scaling as $\mathcal{O}(\sqrt{d^2 \cdot \text{poly}(H) \cdot K})$ would be an interesting direction to pursue. In addition, resolving the optimal H dependence remains an open question. A second interesting direction would be to extend the work of (Tarbouriech et al., 2020) to the linear MDP setting. (Tarbouriech et al., 2020) provides an algorithm that allows the learner to collect an arbitrary number of samples for each state-action pair individually. In contrast, CoverTraj only lets one specify the tolerance for each partition set as whole. The direct generalization of (Tarbouriech et al., 2020) to the linear setting would be to allow the learner to specify the tolerance in each direction individually. We believe CoverTraj could be used as a basic building block in such an approach, but leave this extension for future work.

Acknowledgements

The work of AW is supported by an NSF GFRP Fellowship DGE-1762114. The work of SSD is in part supported by grants NSF IIS-2110170. The work of KJ was funded in part by the AFRL and NSF TRIPODS 2023166.

References

Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24:2312–2320, 2011.

Agarwal, A., Kakade, S., Krishnamurthy, A., and Sun, W. Flambe: Structural complexity and representation learning of low rank mdps. *arXiv preprint arXiv:2006.10814*, 2020.

Agarwal, N., Chaudhuri, S., Jain, P., Nagaraj, D., and Netrapalli, P. Online target q-learning with reverse experience replay: Efficiently finding the optimal policy for linear mdps. *arXiv preprint arXiv:2110.08440*, 2021.

Ayoub, A., Jia, Z., Szepesvari, C., Wang, M., and Yang, L. Model-based reinforcement learning with value-targeted regression. In *International Conference on Machine Learning*, pp. 463–474. PMLR, 2020.

Azar, M. G., Osband, I., and Munos, R. Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, pp. 263–272. PMLR, 2017.

Dann, C. and Brunskill, E. Sample complexity of episodic fixed-horizon reinforcement learning. *arXiv preprint arXiv:1510.08906*, 2015.

Dann, C., Lattimore, T., and Brunskill, E. Unifying pac and regret: Uniform pac bounds for episodic reinforcement learning. *arXiv preprint arXiv:1703.07710*, 2017.

Dann, C., Li, L., Wei, W., and Brunskill, E. Policy certificates: Towards accountable reinforcement learning. In *International Conference on Machine Learning*, pp. 1507–1516. PMLR, 2019.

Du, S. S., Kakade, S. M., Wang, R., and Yang, L. F. Is a good representation sufficient for sample efficient reinforcement learning? *arXiv preprint arXiv:1910.03016*, 2019.

Du, S. S., Kakade, S. M., Lee, J. D., Lovett, S., Mahajan, G., Sun, W., and Wang, R. Bilinear classes: A structural framework for provable generalization in rl. *arXiv preprint arXiv:2103.10897*, 2021.

Foster, D. J., Kakade, S. M., Qian, J., and Rakhlin, A. The statistical complexity of interactive decision making. *arXiv preprint arXiv:2112.13487*, 2021.

Freedman, D. A. On tail probabilities for martingales. *the Annals of Probability*, pp. 100–118, 1975.

Hao, B., Lattimore, T., Szepesvári, C., and Wang, M. Online sparse reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 316–324. PMLR, 2021.

Huang, J., Chen, J., Zhao, L., Qin, T., Jiang, N., and Liu, T.-Y. Towards deployment-efficient reinforcement learning: Lower bound and optimality. *arXiv preprint arXiv:2202.06450*, 2022.

Jia, Z., Yang, L., Szepesvari, C., and Wang, M. Model-based reinforcement learning with value-targeted regression. In *Learning for Dynamics and Control*, pp. 666–686. PMLR, 2020.

Jiang, N., Krishnamurthy, A., Agarwal, A., Langford, J., and Schapire, R. E. Contextual decision processes with low bellman rank are pac-learnable. In *International Conference on Machine Learning*, pp. 1704–1713. PMLR, 2017.

Jin, C., Allen-Zhu, Z., Bubeck, S., and Jordan, M. I. Is q-learning provably efficient? In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 4868–4878, 2018.

Jin, C., Krishnamurthy, A., Simchowitz, M., and Yu, T. Reward-free exploration for reinforcement learning. In *International Conference on Machine Learning*, pp. 4870–4879. PMLR, 2020a.

Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pp. 2137–2143. PMLR, 2020b.

Jin, C., Liu, Q., and Miryoosefi, S. Bellman eluder dimension: New rich classes of rl problems, and sample-efficient algorithms. *arXiv preprint arXiv:2102.00815*, 2021.

Kakade, S. M. *On the sample complexity of reinforcement learning*. PhD thesis, UCL (University College London), 2003.

Kearns, M. and Singh, S. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49(2):209–232, 2002.

Ménard, P., Domingues, O. D., Jonsson, A., Kaufmann, E., Leurent, E., and Valko, M. Fast active learning for pure exploration in reinforcement learning. *arXiv preprint arXiv:2007.13442*, 2020.

Modi, A., Jiang, N., Tewari, A., and Singh, S. Sample complexity of reinforcement learning using linearly combined model ensembles. In *International Conference on Artificial Intelligence and Statistics*, pp. 2010–2020. PMLR, 2020.

Modi, A., Chen, J., Krishnamurthy, A., Jiang, N., and Agarwal, A. Model-free representation learning and exploration in low-rank mdps. *arXiv preprint arXiv:2102.07035*, 2021.

Shamir, O. On the complexity of bandit and derivative-free stochastic convex optimization. In *Conference on Learning Theory*, pp. 3–24. PMLR, 2013.

Tarbouriech, J., Pirotta, M., Valko, M., and Lazaric, A. A provably efficient sample collection strategy for reinforcement learning. *arXiv preprint arXiv:2007.06437*, 2020.

Tsybakov, A. B. Introduction to nonparametric estimation., 2009.

Vershynin, R. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

Wagenmaker, A., Chen, Y., Simchowitz, M., Du, S. S., and Jamieson, K. First-order regret in reinforcement learning with linear function approximation: A robust estimation approach. *arXiv preprint arXiv:2112.03432*, 2021a.

Wagenmaker, A., Simchowitz, M., and Jamieson, K. Beyond no regret: Instance-dependent pac reinforcement learning. *arXiv preprint arXiv:2108.02717*, 2021b.

Wang, R., Du, S. S., Yang, L. F., and Salakhutdinov, R. On reward-free reinforcement learning with linear function approximation. *arXiv preprint arXiv:2006.11274*, 2020.

Wang, Y., Wang, R., Du, S. S., and Krishnamurthy, A. Optimism in reinforcement learning with generalized linear function approximation. *arXiv preprint arXiv:1912.04136*, 2019.

Wang, Y., Wang, R., and Kakade, S. M. An exponential lower bound for linearly-realizable mdps with constant suboptimality gap. *arXiv preprint arXiv:2103.12690*, 2021.

Weisz, G., Amortila, P., and Szepesvári, C. Exponential lower bounds for planning in mdps with linearly-realizable optimal action-value functions. In *Algorithmic Learning Theory*, pp. 1237–1264. PMLR, 2021.

Wu, J., Braverman, V., and Yang, L. F. Gap-dependent unsupervised exploration for reinforcement learning. *arXiv preprint arXiv:2108.05439*, 2021.

Yang, L. and Wang, M. Sample-optimal parametric q-learning using linearly additive features. In *International Conference on Machine Learning*, pp. 6995–7004. PMLR, 2019.

Zanette, A. and Brunskill, E. Tighter problem-dependent regret bounds in reinforcement learning without domain knowledge using value function bounds. In *International Conference on Machine Learning*, pp. 7304–7312. PMLR, 2019.

Zanette, A., Brandfonbrener, D., Brunskill, E., Pirotta, M., and Lazaric, A. Frequentist regret bounds for randomized least-squares value iteration. In *International Conference on Artificial Intelligence and Statistics*, pp. 1954–1964. PMLR, 2020a.

Zanette, A., Lazaric, A., Kochenderfer, M., and Brunskill, E. Learning near optimal policies with low inherent bellman error. In *International Conference on Machine Learning*, pp. 10978–10989. PMLR, 2020b.

Zanette, A., Lazaric, A., Kochenderfer, M. J., and Brunskill, E. Provably efficient reward-agnostic navigation with linear value iteration. *arXiv preprint arXiv:2008.07737*, 2020c.

Zhang, W., Zhou, D., and Gu, Q. Reward-free model-based reinforcement learning with linear function approximation. *Advances in Neural Information Processing Systems*, 34, 2021a.

Zhang, Z., Du, S. S., and Ji, X. Nearly minimax optimal reward-free reinforcement learning. *arXiv preprint arXiv:2010.05901*, 2020a.

Zhang, Z., Ji, X., and Du, S. S. Is reinforcement learning more difficult than bandits? a near-optimal algorithm escaping the curse of horizon. *arXiv preprint arXiv:2009.13503*, 2020b.

Zhang, Z., Yang, J., Ji, X., and Du, S. S. Variance-aware confidence set: Variance-dependent bound for linear bandits and horizon-free bound for linear mixture mdp. *arXiv preprint arXiv:2101.12745*, 2021b.

Zhou, D., Gu, Q., and Szepesvari, C. Nearly minimax optimal reinforcement learning for linear mixture markov decision processes. *arXiv preprint arXiv:2012.08507*, 2020.

Zhou, D., He, J., and Gu, Q. Provably efficient reinforcement learning for discounted mdps with feature mapping. In *International Conference on Machine Learning*, pp. 12793–12802. PMLR, 2021.

A. Technical Results

Definition A.1 (Covering Number)**.** Let $\mathcal{X}$ be a set with metric $\mathrm{dist}(\cdot,\cdot)$. Given $\epsilon > 0$, the ϵ-covering number of $\mathcal{X}$ in dist, $\mathsf{N}(\mathcal{X}, \mathrm{dist}, \epsilon)$, is defined as the minimal cardinality of a set $\mathcal{N} \subset \mathcal{X}$ such that, for all $x \in \mathcal{X}$, there exists an $x' \in \mathcal{N}$ with $\mathrm{dist}(x, x') \le \epsilon$.

Lemma A.1 ((Vershynin, 2010))**.** *For any $\epsilon > 0$, the ϵ-covering number of the Euclidean ball $\mathcal{B}^d(R) := \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_2 = 1\}$ with radius $R > 0$ in the Euclidean metric is upper bounded by $(1 + 2R/\epsilon)^d$.*

Lemma A.2 (Elliptic Potential Lemma, Lemma 11 of (Abbasi-Yadkori et al., 2011))**.** *Consider a sequence of vectors $(\boldsymbol{x}_t)_{t=1}^T, \boldsymbol{x}_t \in \mathbb{R}^d$, and assume that $\|\boldsymbol{x}_t\|_2 \le a$ for all t. Let $\mathbf{V}_t = \lambda I + \sum_{s=1}^t \boldsymbol{x}_s \boldsymbol{x}_s^\top$ for some $\lambda > 0$. Then we will have that*

$$\sum_{t=1}^T \min\{1, \|\boldsymbol{x}_t\|^2_{\mathbf{V}_{t-1}^{-1}}\} \le 2d \log(1 + a^2 T/(d\lambda)).$$

Furthermore, if $\lambda \ge \max\{1, a^2\}$,

$$\sum_{t=1}^T \|\boldsymbol{x}_t\|^2_{\mathbf{V}_{t-1}^{-1}} \le 2d \log(1 + a^2 T/(d\lambda)).$$

Lemma A.3 (Freedman's Inequality (Freedman, 1975))**.** *$\mathcal{F}_0 \subset \mathcal{F}_1 \subset \ldots \subset \mathcal{F}_T$ be a filtration and let $X_1, X_2, \ldots, X_T$ be real random variables such that X_t is $\mathcal{F}_t$-measurable, $\mathbb{E}[X_t|\mathcal{F}_{t-1}] = 0$, $|X_t| \le b$ almost surely, and $\sum_{t=1}^T \mathbb{E}[X_t^2|\mathcal{F}_{t-1}] \le V$ for some fixed $V > 0$ and $b > 0$. Then for any $\delta \in (0,1)$, we have with probability at least $1 - \delta$,*

$$\sum_{t=1}^T X_t \le 2\sqrt{V \log(1/\delta)} + b \log(1/\delta).$$

Lemma A.4. *If $x \ge C(2n)^n \log^n(2nCB)$ for $n, C, B \ge 1$, then $x \ge C \log^n(Bx)$.*

Proof. If $x = C(2n)^n \log^n(2nCB)$, then

$$\begin{aligned}
C \log^n(Bx) &= C \log^n \left[C(2n)^n \log^n(2nCB)\right] \\
&\le C \log^n \left[C^{1+n}(2n)^{2n} B^n\right] \\
&\le C \log^n \left[C^{2n}(2n)^{2n} B^{2n}\right] \\
&\le C(2n)^n \log^n \left[2nCB\right] \\
&= x.
\end{aligned}$$

The result then follows since x increases more quickly than $C \log^n(Bx)$. □

B. Collecting Covering Trajectories

We prove a slightly more general version of Theorem 3. In particular, instead of setting $\mathbf{\Lambda}_{h,k-1} = I + \sum_{\tau=1}^{k-1} \boldsymbol{\phi}_{h,\tau} \boldsymbol{\phi}_{h,\tau}^\top$ as in EGS, we prove the result for the setting of $\mathbf{\Lambda}_{h,k-1} = \lambda I + \sum_{\tau=1}^{k-1} \boldsymbol{\phi}_{h,\tau} \boldsymbol{\phi}_{h,\tau}^\top$ in EGS, for some $\lambda > 0$ (for convenience we also assume $\lambda \le \max\{\mathcal{C}_1, \mathcal{C}_2, 1\}$, though this can be relaxed if desired). For the proof of Theorem 1 we simply set $\lambda = 1$.

In COVERTRAJ, the precise setting of K_i is

$$K_i \leftarrow \left\lceil 2^i \cdot \max\left\{ 2^{10+p_1} \mathcal{C}_1 p_1^{p_1} \log^{p_1}\left(\frac{2^{i+12} p_1 m \mathcal{C}_1 H}{\delta}\right), 2^{4+p_2} \mathcal{C}_2 p_2^{p_2} \log^{p_2}\left(\frac{2^{i+6} p_2 m \mathcal{C}_2 H}{\delta}\right), \frac{24d}{\boldsymbol{\gamma}_i^2} \log \frac{48 \cdot 2^i d/\lambda}{\boldsymbol{\gamma}_i^2} \right\} \right\rceil.$$

Proof of Theorem 3. Theorem 3 is a direct consequence of Lemma B.1. For $i = 1$, it is clearly the case that

$$\sup_\pi w_h^\pi(\mathcal{X}) \le 2^{-i+1} = 1,$$

so Lemma B.1 and our choice of K_1 gives that with probability at least $1-\delta/m$,

$$\sup_\pi w_h^\pi(\mathcal{X}\backslash\mathcal{X}_1) \le 2^{-i} \quad \text{and} \quad \boldsymbol{\phi}^\top \boldsymbol{\Lambda}_1^{-1}\boldsymbol{\phi} \le \boldsymbol{\gamma}_1^2, \forall \boldsymbol{\phi}\in\mathcal{X}_1.$$

Now assume that for some i,

$$\sup_\pi w_h^\pi(\mathcal{X}) \le 2^{-i+1},$$

then again Lemma B.1 and our choice of K_i gives that with probability at least $1-\delta/m$,

$$\sup_\pi w_h^\pi(\mathcal{X}\backslash\mathcal{X}_i) \le 2^{-i} \quad \text{and} \quad \boldsymbol{\phi}^\top \boldsymbol{\Lambda}_i^{-1}\boldsymbol{\phi} \le \boldsymbol{\gamma}_i^2, \forall \boldsymbol{\phi}\in\mathcal{X}_i.$$

The result follows by union bounding over the success event of Lemma B.1 holding for each $i\in[m]$. The final conclusion holds since after epoch m, $\mathcal{X} = \mathcal{B}^d\backslash \cup_{i=1}^m \mathcal{X}_i$. □

Lemma B.1. *Consider running Algorithm 3 with $\gamma\in(0,1]$ and some regret-minimization algorithm* RegMin *satisfying Definition 5.1, and with input set $\mathcal{X}$ satisfying*

$$\sup_\pi \omega_h^\pi(\mathcal{X}) \le 2^{-i}.$$

Assume also that K is chosen to satisfy

$$K \ge \left\lceil 2^i \cdot \max\Big\{ 1024\mathcal{C}_1\cdot(2p_1)^{p_1}\log^{p_1}\cdot[4096\cdot 2^i p_1\mathcal{C}_1 H/\delta], \\ 16\mathcal{C}_2\cdot(2p_2)^{p_2}\log^{p_2}\cdot[64\cdot 2^i p_2\mathcal{C}_2 H/\delta], \frac{24d}{\gamma^2}\log\frac{48\cdot 2^i d}{\gamma^2}\Big\}\right\rceil. \tag{B.1}$$

Let $\widetilde{\mathcal{X}}\subseteq\mathbb{R}^d$ denote the set returned by Algorithm 3 defined as

$$\widetilde{\mathcal{X}} = \{\boldsymbol{\phi}\in\mathcal{X} \,:\, \boldsymbol{\phi}^\top\boldsymbol{\Lambda}_{h,K}^{-1}\boldsymbol{\phi}\le\gamma^2\}.$$

Then, with probability at least $1-\delta$,

$$\sup_\pi \omega_h^\pi(\mathcal{X}\backslash\widetilde{\mathcal{X}}) \le 2^{-i-1}.$$

Proof. First note that the reward sequence used in Algorithm 3 satisfies the conditions of Definition 5.1. Thus, we have that, with probability at least $1-\delta$,

$$\sum_{k=1}^K [V_0^\star(r^k) - V_0^{\pi_k}(r^k)] \le \sqrt{\mathcal{C}_1 V_0^\star(r^1) K\cdot\log^{p_1}(HK/\delta)} + \mathcal{C}_2\log^{p_2}(HK/\delta). \tag{B.2}$$

For simplicity we will assume that $\mathcal{C}_1,\mathcal{C}_2,p_1,p_2\ge 1$ (since if this is not true, for example if $\mathcal{C}_1<1$, (B.2) still holds with $\mathcal{C}_1$ replaced by $\max\{\mathcal{C}_1,1\}$).

Relating $\sum_{k=1}^K V_0^{\pi_k}(r^k)$ to random reward. Note that $r_h^k(s,a)\le 1$ for all s,a, and since the reward is non-zero only at step h, $\mathbb{E}_{\pi_k}[r_h^k(s_h^k,a_h^k)] = V_0^{\pi_k}(r^k)$ and $V_0^{\pi_k}(r_k)\le 1$. Then, since the reward is non-increasing,

$$\mathbb{E}[(r_h^k(s_h^k,a_h^k) - V_0^{\pi_k}(r^k))^2|\mathcal{F}_{k-1}] \le 2V_0^{\pi_k}(r^k) \le 2V_0^\star(r^1).$$

By Freedman's inequality, Lemma A.3, it follows that with probability at least $1-\delta$,

$$\left|\sum_{k=1}^K [r_h^k(s_h^k,a_h^k) - V_0^{\pi_k}(r^k)]\right| \le \sqrt{8V_0^\star(r^1)K\log 1/\delta} + \log 1/\delta.$$

Thus, union bounding over this event and the event of (B.2), we have with probability $1-\delta$ that

$$\begin{aligned}\sum_{k=1}^{K} r_h^k(s_h^k, a_h^k) &\geq \sum_{k=1}^{K} V_0^{\pi_k}(r^k) - \sqrt{8V_0^\star(r^1)K\log 2/\delta} - \log 2/\delta \\ &\geq \sum_{k=1}^{K} V_0^\star(r^k) - \sqrt{16\mathcal{C}_1 V_0^\star(r^1)K \cdot \log^{p_1}(2HK/\delta)} - 2\mathcal{C}_2 \cdot \log^{p_2}(2HK/\delta) \end{aligned} \tag{B.3}$$

where we have used that $\mathcal{C}_1, \mathcal{C}_2, p_1, p_2 \geq 1$ to group terms.

Proof by contradiction. Our goal is use (B.4) to reach a contradiction and show that, for our choice of K, $V_0^\star(r^K) \leq 2^{-i-1}$. To set up the argument, assume for the sake of contradiction that $V_0^\star(r^K) > 2^{-i-1}$. Since r^k is non-increasing, this implies that $V_0^\star(r^k) > 2^{-i-1}$ for all $k \in [K]$. Then we can lower bound (B.3) as

$$\text{(B.3)} > K2^{-i-1} - \sqrt{16\mathcal{C}_1 V_0^\star(r^1)K \cdot \log^{p_1}(2HK/\delta)} - 2\mathcal{C}_2 \cdot \log^{p_2}(2HK/\delta). \tag{B.4}$$

Upper Bounding the Reward. We next upper bound the total reward that can be obtained:

$$\begin{aligned}\sum_{k=1}^{K} r_h^k(s_h^k, a_h^k) &= \sum_{k=1}^{K} \min\{1, \gamma^{-2}\|\boldsymbol{\phi}(s_h^k, a_h^k)\|^2_{\boldsymbol{\Lambda}_{h,k-1}^{-1}}\} \cdot \mathbb{I}\{\boldsymbol{\phi}(s_h^k, a_h^k) \in \mathcal{X}\} \\ &\leq \frac{1}{\gamma^2} \cdot \sum_{k=1}^{K} \min\{1, \|\boldsymbol{\phi}(s_h^k, a_h^k)\|^2_{\boldsymbol{\Lambda}_{h,k-1}^{-1}}\} \cdot \mathbb{I}\{\boldsymbol{\phi}(s_h^k, a_h^k) \in \mathcal{X}\} \\ &\leq \frac{1}{\gamma^2} \cdot \sum_{k=1}^{K} \min\{1, \|\boldsymbol{\phi}(s_h^k, a_h^k)\|^2_{\boldsymbol{\Lambda}_{h,k-1}^{-1}}\} \\ &\leq \frac{1}{\gamma^2} \cdot 2d\log(1 + K/d\lambda) \end{aligned} \tag{B.5}$$

where we have used that $\gamma \leq 1$, and where the last inequality follows by the Elliptic Potential Lemma, Lemma A.2, since $\|\boldsymbol{\phi}(s_h^k, a_h^k)\|_2 \leq 1$ by assumption, and we normalize $\boldsymbol{\Lambda}_{h,k}$ by λI.

Lower Bounding the Reward. By assumption, we have that $\sup_\pi \omega_h^\pi(\mathcal{X}) \leq 2^{-i}$. This implies that

$$V_0^\star(r^1) = \sup_\pi \mathbb{E}_\pi[r_h^1(s_h, a_h)] \leq \sup_\pi \mathbb{E}_\pi[\mathbb{I}\{\boldsymbol{\phi}(s_h, a_h) \in \mathcal{X}\}] \leq 2^{-i}$$

where the last inequality follows since $\sup_\pi \mathbb{E}_\pi[\mathbb{I}\{\boldsymbol{\phi}(s_h, a_h) \in \mathcal{X}\}] = \sup_\pi \omega_h^\pi(\mathcal{X})$ by definition. Then,

$$\text{(B.4)} \geq K2^{-i-1} - \sqrt{2^{-i} \cdot 16\mathcal{C}_1 K \cdot \log^{p_1}(2HK/\delta)} - 2\mathcal{C}_2 \cdot \log^{p_2}(2HK/\delta). \tag{B.6}$$

Assume that K is chosen such that

$$K \geq 1024 \cdot 2^i \mathcal{C}_1 (2p_1)^{p_1} \log^{p_1}[4096 \cdot 2^i p_1 \mathcal{C}_1 H/\delta] \tag{B.7}$$

then by Lemma A.4 we will have

$$\frac{1}{4}K2^{-i-1} - \sqrt{2^{-i} \cdot 16\mathcal{C}_1 K \cdot \log^{p_1}(2HK/\delta)} \geq 0.$$

Similarly, if

$$K \geq 16 \cdot 2^i \mathcal{C}_2 (2p_2)^{p_2} \log^{p_2}[64 \cdot 2^i p_2 \mathcal{C}_2 H/\delta] \tag{B.8}$$

then

$$\frac{1}{4}K2^{-i-1} - 2\mathcal{C}_2 \cdot \log^{p_2}(2HK/\delta) \geq 0.$$

As (B.1) requires that K is chosen so as to satisfy both (B.7) and (B.8), it follows that

$$(\text{B.6}) \geq \frac{1}{2} K 2^{-i-1}.$$

However, (B.1) also gives that $K \geq \frac{2^i \cdot 24d}{\gamma^2} \log \frac{2^i \cdot 48d/\lambda}{\gamma^2}$. Lemma A.4 then implies that

$$K \geq \frac{2^i \cdot 12d}{\gamma^2} \log(2K/\lambda)$$

so that, stringing together the above inequalities,

$$\sum_{k=1}^{K} r_h^k(s_h^k, a_h^k) \geq (\text{B.3}) > (\text{B.4}) \geq (\text{B.6}) \geq \frac{1}{2} K 2^{-i-1} \geq \frac{3d}{\gamma^2} \log(2K/\lambda). \tag{B.9}$$

Concluding the proof. Combining Equations (B.5) and (B.9), we have shown that

$$\frac{2d}{\gamma^2} \log(1 + K/d\lambda) \geq \sum_{k=1}^{K} r_h^k(s_h^k, a_h^k) > (\text{B.6}) \geq \frac{3d}{\gamma^2} \log(2K/\lambda).$$

This is a contradiction, since $\frac{2d}{\gamma^2} \log(1 + K/d\lambda) \leq \frac{3d}{\gamma^2} \log(2K/\lambda)$. Thus, with probability $1 - \delta$, we must have that $V_1^\star(r^K) \leq 2^{-i-1}$. The conclusion that $\sup_\pi \omega_h^\pi(\mathcal{X} \backslash \widetilde{\mathcal{X}}) \leq 2^{-i-1}$ follows on this event since

$$\begin{aligned}
V_0^\star(r^K) &= \sup_\pi \mathbb{E}_\pi[\mathbb{I}\{\boldsymbol{\phi}(s_h, a_h)^\top \boldsymbol{\Lambda}_{h,K-1}^{-1} \boldsymbol{\phi}(s_h, a_h) > \gamma^2, \boldsymbol{\phi}(s_h, a_h) \in \mathcal{X}\}] \\
&\quad + \mathbb{E}_\pi[\gamma^{-2} \boldsymbol{\phi}(s_h, a_h)^\top \boldsymbol{\Lambda}_{h,K-1}^{-1} \boldsymbol{\phi}(s_h, a_h) \cdot \mathbb{I}\{\boldsymbol{\phi}(s_h, a_h)^\top \boldsymbol{\Lambda}_{h,K-1}^{-1} \boldsymbol{\phi}(s_h, a_h) \leq \gamma^2, \boldsymbol{\phi}(s_h, a_h) \in \mathcal{X}\}] \\
&\geq \sup_\pi \mathbb{E}_\pi[\mathbb{I}\{\boldsymbol{\phi}(s_h, a_h)^\top \boldsymbol{\Lambda}_{h,K}^{-1} \boldsymbol{\phi}(s_h, a_h) > \gamma^2, \boldsymbol{\phi}(s_h, a_h) \in \mathcal{X}\}] \\
&= \sup_\pi \mathbb{E}_\pi[\mathbb{I}\{\boldsymbol{\phi}(s_h, a_h) \in \mathcal{X} \backslash \widetilde{\mathcal{X}}\}]
\end{aligned}$$

where the final equality follows by definition of $\widetilde{\mathcal{X}}$.

□

B.1. FORCE satisfies Definition 5.1

We invoke Theorem 8 of (Wagenmaker et al., 2021a). To do so, we need to control an appropriate covering number. Recall the ball $\mathcal{B}^d := \{\boldsymbol{\phi} \in \mathbb{R}^d : \|\boldsymbol{\phi}\| \leq 1\}$. Given a class of functions $\mathscr{F}$ denote a set of functions $f : \mathcal{B}^d \to \mathbb{R}$, we define the distance on $f_1, f_2 \in \mathscr{F}$

$$\text{dist}_\infty(f_1, f_2) := \sup_{\boldsymbol{\phi} \in \mathcal{B}^d} |f_1(\boldsymbol{\phi}) - f_2(\boldsymbol{\phi})|.$$

Lemma B.2. *Consider the class of functions*

$$\mathscr{R} := \left\{ r : \mathcal{B}^d \to \mathbb{R} \ : \ r(\boldsymbol{\phi}) = \begin{cases} 1 & \|\boldsymbol{\phi}\|_{\boldsymbol{\Lambda}^{-1}}^2 > \gamma^2, \boldsymbol{\phi} \in \mathcal{X} \\ \gamma^{-2} \|\boldsymbol{\phi}\|_{\boldsymbol{\Lambda}^{-1}}^2 & \|\boldsymbol{\phi}\|_{\boldsymbol{\Lambda}^{-1}}^2 \leq \gamma^2, \boldsymbol{\phi} \in \mathcal{X} \\ 0 & \boldsymbol{\phi} \notin \mathcal{X} \end{cases}, \boldsymbol{\Lambda} \succeq I \right\}$$

for some $\gamma^2 > 0$ and $\mathcal{X} \subseteq \mathbb{R}^d$. Then

$$\mathsf{N}(\mathscr{R}, \text{dist}_\infty, \epsilon) \leq d^2 \log\left(1 + \frac{2\sqrt{d}}{\gamma^2 \epsilon}\right).$$

Proof. Consider $r_1, r_2 \in \mathscr{R}$ parameterized by $\mathbf{\Lambda}_1, \mathbf{\Lambda}_2$. Then,

$$\mathrm{dist}_\infty(r_1, r_2) = \sup_{\boldsymbol{\phi} \in \mathcal{B}^d} |r_1(\boldsymbol{\phi}) - r_2(\boldsymbol{\phi})| \leq \sup_{\boldsymbol{\phi} \in \mathcal{B}^d} \frac{1}{\gamma^2} \left| \|\boldsymbol{\phi}\|^2_{\mathbf{\Lambda}_1^{-1}} - \|\boldsymbol{\phi}\|^2_{\mathbf{\Lambda}_2^{-1}} \right|.$$

The inequality follows easily by noting that in every possible case, we can bound

$$|r_1(\boldsymbol{\phi}) - r_2(\boldsymbol{\phi})| \leq \frac{1}{\gamma^2} \left| \|\boldsymbol{\phi}\|^2_{\mathbf{\Lambda}_1^{-1}} - \|\boldsymbol{\phi}\|^2_{\mathbf{\Lambda}_2^{-1}} \right|.$$

Now note that

$$\left| \|\boldsymbol{\phi}\|^2_{\mathbf{\Lambda}_1^{-1}} - \|\boldsymbol{\phi}\|^2_{\mathbf{\Lambda}_2^{-1}} \right| = \left| \boldsymbol{\phi}^\top (\mathbf{\Lambda}_1^{-1} - \mathbf{\Lambda}_2^{-1}) \boldsymbol{\phi} \right| \leq \|\mathbf{\Lambda}_1^{-1} - \mathbf{\Lambda}_2^{-1}\|_{\mathrm{op}} \leq \|\mathbf{\Lambda}_1^{-1} - \mathbf{\Lambda}_2^{-1}\|_{\mathrm{F}}.$$

Let $\mathcal{N}$ be an $\gamma^2\epsilon$ cover of $\{\mathbf{A} \in \mathbb{R}^{d \times d} \ : \ \|\mathbf{A}\|_{\mathrm{F}} \leq \sqrt{d}\}$. Then by Lemma A.1, $\log|\mathcal{N}| \leq d^2 \log(1 + 2\sqrt{d}/(\gamma^2\epsilon))$. Furthermore, for any $\mathbf{\Lambda} \succeq I$, we can find some $\mathbf{A} \in \mathcal{N}$ such that $\|\mathbf{\Lambda}^{-1} - \mathbf{A}\|_{\mathrm{F}} \leq \gamma^2\epsilon$. Let

$$\mathscr{V} := \left\{ r(\cdot) \ : \ r(\boldsymbol{\phi}) = \begin{cases} 1 & \|\boldsymbol{\phi}\|^2_{\mathbf{A}^{-1}} > \gamma^2, \boldsymbol{\phi} \in \mathcal{X} \\ \gamma^{-2}\|\boldsymbol{\phi}\|^2_{\mathbf{A}} & \|\boldsymbol{\phi}\|^2_{\mathbf{A}} \leq \gamma^2, \boldsymbol{\phi} \in \mathcal{X} \\ 0 & \boldsymbol{\phi} \notin \mathcal{X} \end{cases}, \mathbf{A} \in \mathcal{N} \right\},$$

then it follows that $\mathscr{V}$ is an ϵ-net of $\mathscr{R}$ in the dist_∞ norm, and that $\log|\mathscr{V}| \leq d^2 \log(1+2\sqrt{d}/(\gamma^2\epsilon))$. The result follows. □

Lemma B.3. *Let r^k be as defined in* EGS. *Then* FORCE *satisfies Definition 5.1 with*

$$\mathcal{C}_1 = d^4 H^3 \log(e + \sqrt{d}/\gamma^2), \quad p_1 = 3$$
$$\mathcal{C}_2 = d^4 H^3 \log^{3/2}(e + \sqrt{d}/\gamma^2), \quad p_2 = 7/2$$

Proof. This follows directly from Theorem 8 of (Wagenmaker et al., 2021a) by noting that r_h^k is non-increasing in k, $\mathcal{F}_{k-1}$-measurable, $r_h^k \in \mathscr{R}$, and using the covering number for $\mathscr{R}$ given in Lemma B.2. □

C. Minimax Optimal Reward-Free RL

Let us first establish notation. Recall the episode magnitudes K_i from Algorithm 4. We define

$$K_{\mathrm{tot}} := \sum_{i=1}^{\iota_\epsilon} K_i, \quad \text{where } \iota_\epsilon := \lceil \log_2(\frac{\epsilon}{4\beta H}) \rceil$$

We assume that our parameters $K_{\max}$ and β are chosen such that

$$\beta \geq \widetilde{\beta} := cH\sqrt{d \log(1 + dHK_{\mathrm{tot}}) + \log H/\delta}, \quad K_{\max} \geq K_{\mathrm{tot}}.$$

Throughout this section we will consider an arbitrary reward function satisfying Definition 3.1, and will let $V^\star, V^\pi, Q^\star, Q^\pi$ denote the value functions for $\pi^\star$ and π, respectively, with respect to r. Similarly, we will let V and Q refer to the value function estimates maintained by RFLIN-PLAN when run with r as an input.

Proof of Theorem 1. First we establish ϵ-suboptimality, then we address sample complexity.

ϵ-suboptimality. Fix some reward function r satisfying Definition 3.1. Let $\widehat{\pi}$ denote the policy returned by RFLIN-PLAN(r). As noted above, let $V^\star, V^\pi, V, Q^\star, Q^\pi, Q$ refer to the value functions with respect to r.

Let $\mathcal{E}_Q$ denote the high-probability even from Lemma C.4, which holds with probability at least $1 - \delta$. By Lemma C.6 and the choice of parameter $\beta \geq \widetilde{\beta}$, the following holds on $\mathcal{E}_Q$:

$$V_0^\star - V_0^{\widehat{\pi}} \leq V_0 - V_0^{\widehat{\pi}}.$$

Moreover, by Lemma C.5 and, again using $\beta \geq \widetilde{\beta}$, we have

$$|\langle \boldsymbol{\phi}(s,a), \widehat{\boldsymbol{w}}_h \rangle - r_h(s,a) - \mathbb{E}_h[V_{h+1}](s,a)| \leq \widetilde{\beta} \|\boldsymbol{\phi}(s,a)\|_{\boldsymbol{\Lambda}_h^{-1}} \leq \beta \|\boldsymbol{\phi}(s,a)\|_{\boldsymbol{\Lambda}_h^{-1}}.$$

Thus, by definition of Q, we have

$$\begin{aligned} V_h(s_h) = Q_h(s_h, \widehat{\pi}_h(s_h)) &\leq \langle \boldsymbol{\phi}(s_h, \widehat{\pi}_h(s_h)), \widehat{\boldsymbol{w}}_h \rangle + \beta \|\boldsymbol{\phi}(s_h, \widehat{\pi}_h(s_h))\|_{\boldsymbol{\Lambda}_h^{-1}} \\ &\leq r_h(s_h, \widehat{\pi}_h(s_h)) + \mathbb{E}_h[V_{h+1}](s_h, \widehat{\pi}_h(s_h)) + 2\beta \|\boldsymbol{\phi}(s_h, \widehat{\pi}_h(s_h))\|_{\boldsymbol{\Lambda}_h^{-1}}. \end{aligned}$$

Similarly, by the Bellman Equation, we have

$$V_h^{\widehat{\pi}}(s_h) = Q_h^{\widehat{\pi}}(s_h, \widehat{\pi}_h(s_h)) = r_h(s_h, \widehat{\pi}_h(s_h)) + \mathbb{E}_h[V_{h+1}^{\widehat{\pi}}](s_h, \widehat{\pi}_h(s_h)).$$

It follows that

$$V_h(s_h) - V_h^{\widehat{\pi}}(s_h) \leq \mathbb{E}_h[V_{h+1} - V_{h+1}^{\widehat{\pi}}](s_h, \widehat{\pi}_h(s_h)) + 2\beta \|\boldsymbol{\phi}(s_h, \widehat{\pi}_h(s_h))\|_{\boldsymbol{\Lambda}_h^{-1}}.$$

Thus, unrolling this backwards gives (since $\widehat{\pi}$ is deterministic):

$$\begin{aligned} V_0 - V_0^{\widehat{\pi}} &\leq \mathbb{E}_1[V_2 - V_2^{\widehat{\pi}}](s_1, \widehat{\pi}_1(s_1)) + 2\beta \|\boldsymbol{\phi}(s_1, \widehat{\pi}_1(s_1))\|_{\boldsymbol{\Lambda}_1^{-1}} \\ &\leq \mathbb{E}_1[\mathbb{E}_2[V_3 - V_3^{\widehat{\pi}}](s_2, \widehat{\pi}_2(s_2))](s_1, \widehat{\pi}_1(s_1)) + 2\beta \mathbb{E}_1[\|\boldsymbol{\phi}(s_2, \widehat{\pi}_2(s_2))\|_{\boldsymbol{\Lambda}_2^{-1}}](s_1, \widehat{\pi}_1(s_1)) \\ &\qquad + 2\beta \|\boldsymbol{\phi}(s_1, \widehat{\pi}_1(s_1))\|_{\boldsymbol{\Lambda}_1^{-1}} \\ &= \mathbb{E}_{\widehat{\pi}}[V_3(s_3) - V_3^{\widehat{\pi}}(s_3)] + 2\beta \sum_{h=1}^{2} \mathbb{E}_{\widehat{\pi}}[\|\boldsymbol{\phi}(s_h, \widehat{\pi}_h(s_h))\|_{\boldsymbol{\Lambda}_h^{-1}}] \\ &\vdots \\ &\leq 2\beta \sum_{h=1}^{H} \mathbb{E}_{\widehat{\pi}}[\|\boldsymbol{\phi}(s_h, \widehat{\pi}_h(s_h))\|_{\boldsymbol{\Lambda}_h^{-1}}]. \end{aligned}$$

We then upper bound

$$\begin{aligned} 2\beta \sum_{h=1}^{H} \mathbb{E}_{\widehat{\pi}}[\|\boldsymbol{\phi}(s_h, \widehat{\pi}_h(s_h))\|_{\boldsymbol{\Lambda}_h^{-1}}] &\leq 2\beta \sum_{h=1}^{H} \sup_{\pi} \mathbb{E}_{\pi}[\|\boldsymbol{\phi}(s_h, a_h)\|_{\boldsymbol{\Lambda}_h^{-1}}] \\ &\leq 2\beta \sum_{h=1}^{H} \sum_{i=1}^{\iota_\epsilon + 1} \sup_{\pi} \mathbb{E}_{\pi}[\|\boldsymbol{\phi}(s_h, a_h)\|_{\boldsymbol{\Lambda}_h^{-1}} \cdot \mathbb{I}\{\boldsymbol{\phi}(s_h, a_h) \in \mathcal{X}_{h,i}\}] \\ &\leq 2\beta \sum_{h=1}^{H} \sum_{i=1}^{\iota_\epsilon + 1} \sup_{\boldsymbol{\phi} \in \mathcal{X}_{h,i}} \|\boldsymbol{\phi}\|_{\boldsymbol{\Lambda}_h^{-1}} \cdot \sup_{\pi} \mathbb{E}_{\pi}[\mathbb{I}\{\boldsymbol{\phi}(s_h, a_h) \in \mathcal{X}_{h,i}\}]. \end{aligned}$$

By Theorem 3 and a union bound over H, we will have that, with probability at least $1 - \delta$, for all $h \in [H]$ and $i \in [\iota_\epsilon]$ simultaneously,

$$\sup_{\pi} \mathbb{E}_{\pi}[\mathbb{I}\{\boldsymbol{\phi}(s_h, a_h) \in \mathcal{X}_{h,i}\}] = \sup_{\pi} \omega_h^{\pi}(\mathcal{X}_{h,i}) \leq 2^{-i+1}.$$

Furthermore, Theorem 3 also gives

$$\sup_{\boldsymbol{\phi} \in \mathcal{X}_{h,i}} \|\boldsymbol{\phi}\|_{\boldsymbol{\Lambda}_h^{-1}} \leq \sup_{\boldsymbol{\phi} \in \mathcal{X}_{h,i}} \|\boldsymbol{\phi}\|_{\boldsymbol{\Lambda}_{h,i}^{-1}} \leq \sqrt{\boldsymbol{\gamma}_i^2} = \frac{2^i \epsilon}{8H\iota_\epsilon \beta}$$

where the final equality follows by our setting of $\boldsymbol{\gamma}_i^2 = \frac{2^{2i}\epsilon^2}{64H^2\iota_\epsilon^2\beta^2}$. Finally, one last invocation of Theorem 3, followed by the choice of ι_ϵ, gives that

$$\sup_{\pi} \mathbb{E}_{\pi}[\mathbb{I}\{\boldsymbol{\phi}(s_h, a_h) \in \mathcal{X}_{h,\iota_\epsilon+1}\}] = \sup_{\pi} \omega_h^{\pi}(\mathcal{X}_{h,\iota_\epsilon+1}) \leq 2^{-\iota_\epsilon} \leq \frac{\epsilon}{4\beta H}$$

Lastly, observe that $\sup_{\boldsymbol{\phi} \in \mathcal{X}_{h,\iota_\epsilon+1}} \|\boldsymbol{\phi}\|_{\boldsymbol{\Lambda}_h^{-1}} \le 1$ always holds, since $\boldsymbol{\Lambda}_h \succeq I$ and $\|\boldsymbol{\phi}\|_2 \le 1$. This gives that

$$\begin{aligned} 2\beta \sum_{h=1}^{H} \sum_{i=1}^{\iota_\epsilon+1} \sup_{\boldsymbol{\phi} \in \mathcal{X}_{h,i}} \|\boldsymbol{\phi}\|_{\boldsymbol{\Lambda}_h^{-1}} \cdot \sup_{\pi} \mathbb{E}_\pi[\mathbb{I}\{\boldsymbol{\phi}(s_h, a_h) \in \mathcal{X}_{h,i}\}] &\le 2\beta \sum_{h=1}^{H} \sum_{i=1}^{\iota_\epsilon} 2^{-i+1} \frac{2^i \epsilon}{8H\iota_\epsilon \beta} + \frac{\epsilon}{2} \\ &= \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon \end{aligned}$$

so our policy $\widehat{\pi}$ is ϵ-optimal for the reward function r. However, since r was arbitrary, the above holds for all r satisfying Definition 3.1.

Sample complexity. It remains to bound the sample complexity. We note that we ran for a total of $\sum_{h=1}^{H} \sum_{i=1}^{\iota_\epsilon} K_i = HK_{\mathrm{tot}}$ episodes, where again we recall

$$\begin{aligned} K_i = \Bigg\lceil 2^i \cdot \max \Big\{ &1024\mathcal{C}_1 (2p_1)^{p_1} \log^{p_1}[4096 \cdot 2^i p_1 \iota_\epsilon \mathcal{C}_1 H^2/\delta], \\ &16\mathcal{C}_2 (2p_2)^{p_2} \log^{p_2}[64 \cdot 2^i p_2 \iota_\epsilon \mathcal{C}_2 H^2/\delta], \frac{24d}{\boldsymbol{\gamma}_i^2} \log \frac{48 \cdot 2^i d}{\boldsymbol{\gamma}_i^2} \Big\} \Bigg\rceil. \end{aligned}$$

As we use FORCE as our regret minimization algorithm, using the values of $\mathcal{C}_1, \mathcal{C}_2, p_1, p_2$ given in Lemma B.3 we can bound (for a universal constant c)

$$\begin{aligned} K_i &\le c 2^i d^4 H^3 \log^{3/2}(\sqrt{d}/\boldsymbol{\gamma}_i^2) i^3 \log^{7/2}\left(\iota_\epsilon dH \log^{3/2}(\sqrt{d}/\boldsymbol{\gamma}_i^2)/\delta\right) + 2^i \frac{24d}{\boldsymbol{\gamma}_i^2} \log \frac{48 \cdot 2^i d}{\boldsymbol{\gamma}_i^2} \\ &\le 2^i d^4 H^3 \log^{7/2}(1/\delta) \cdot \mathrm{poly}\log(d, H, 1/\epsilon, \log 1/\delta) + \frac{2^i d}{\boldsymbol{\gamma}_i^2} \cdot \mathrm{poly}\log(d, H, 1/\epsilon, \log 1/\delta) \end{aligned}$$

where the last inequality follows by our choice of $\boldsymbol{\gamma}_i^2 = 2^{2i} \cdot \frac{\epsilon^2}{64 H^2 \iota_\epsilon^2 \beta^2}$ and $\iota_\epsilon = \lceil \log_2(4\beta H/\epsilon) \rceil$, and upper bounding $\mathrm{poly}(i)$ factors by $\mathrm{poly}(\iota_\epsilon)$. Let $\iota = \mathrm{poly}\log(d, H, 1/\epsilon, \log 1/\delta)$ (the precise setting of which may change from line to line). Then we can bound the sample complexity by:

$$\begin{aligned} \sum_{h=1}^{H} \sum_{i=1}^{\iota_\epsilon} K_i &\le \iota H \sum_{i=1}^{\iota_\epsilon} \left(2^i d^4 H^3 \log^{7/2}(1/\delta) + \frac{2^i d}{\boldsymbol{\gamma}_i^2} \right) \\ &\le \iota H \sum_{i=1}^{\iota_\epsilon} \left(2^i d^4 H^3 \log^{7/2}(1/\delta) + \frac{2^i d H^2 \beta^2}{2^{2i} \epsilon^2} \right) \\ &\le \frac{d^4 H^5 \beta \log^{7/2}(1/\delta) \iota}{\epsilon} + \frac{d H^3 \beta^2 \iota}{\epsilon^2} \\ &\le \frac{d^{9/2} H^6 \log^4(1/\delta) \iota}{\epsilon} + \frac{d H^5 (d + \log 1/\delta) \iota}{\epsilon^2}. \end{aligned}$$

Selecting β. It remains to show that $K_{\max} \ge K_{\mathrm{tot}}$, which will imply that our setting of β in RFLIN-PLAN satisfies $\beta \ge \widetilde{\beta}$. However, this follows directly by the above complexity bound, which is polynomial in all arguments, justifying our choice of $K_{\max}$.

□

C.1. Concentration of Least Squares Estimates

Lemma C.1 (Lemma B.2 of (Jin et al., 2020b))**.** *When running* RFLIN-PLAN*, for all h,*

$$\|\widehat{\boldsymbol{w}}_h\|_2 \le 2H\sqrt{dK_{\mathrm{tot}}}.$$

Proof. Note that our construction of $\widehat{\boldsymbol{w}}_h$ is identical to the construction of $\boldsymbol{w}_h^K$ in (Jin et al., 2020b), so we can apply Lemma B.2 of (Jin et al., 2020b). □

Lemma C.2. *Consider the function class*

$$\mathscr{F}(\mathbf{\Lambda}, \alpha) := \left\{ V(\cdot) = \min\left\{ \max_{a \in \mathcal{A}} \langle \boldsymbol{w}, \boldsymbol{\phi}(\cdot, a) \rangle + \widetilde{\beta} \|\boldsymbol{\phi}(\cdot, a)\|_{\mathbf{\Lambda}^{-1}}, H \right\}, \|\boldsymbol{w}\|_2 \leq \alpha \right\}$$

for some fixed $\Lambda \succ 0$ and $\widetilde{\beta} > 0$, $\alpha > 0$. Then

$$\mathsf{N}(\mathscr{F}(\mathbf{\Lambda}, \alpha), \mathrm{dist}_\infty, \epsilon) \leq d \log(1 + 2\alpha/\epsilon).$$

Proof. Take some $V_1, V_2 \in \mathscr{F}(\mathbf{\Lambda}, \alpha)$. Since both $\min\{\cdot, H\}$ and $\max_a$ are contraction maps, and $\boldsymbol{\phi}(s, a) \in \mathcal{B}^d$ for all s, a,

$$\sup_s |V_1(s) - V_2(s)| \leq \sup_{\boldsymbol{\phi} \in \mathcal{B}^d} |\langle \boldsymbol{w}_1 - \boldsymbol{w}_2, \boldsymbol{\phi} \rangle| \leq \|\boldsymbol{w}_1 - \boldsymbol{w}_2\|_2.$$

Let $\mathcal{N}$ denote an ϵ-cover of the α-ball, $\mathcal{B}^d(\alpha)$. By Lemma A.1, $\log |\mathcal{N}| \leq d \log(1 + 2\alpha/\epsilon)$. By the above, we then have that for any $V \in \mathscr{F}(\mathbf{\Lambda}, \alpha)$, there exists some $V' \in \mathcal{N}$ such that $\mathrm{dist}_\infty(V, V') \leq \epsilon$, which completes the proof. □

Lemma C.3 (Lemma D.4 of (Jin et al., 2020b))**.** *Let $\{s_\tau\}_{\tau=1}^\infty$ be a stochastic process on state space $\mathcal{S}$ with corresponding filtration $\{\mathcal{F}_\tau\}_{\tau=0}^\infty$. Let $\{\boldsymbol{\phi}_\tau\}_{\tau=0}^\infty$ be an $\mathbb{R}^d$-valued stochastic process where $\boldsymbol{\phi}_\tau \in \mathcal{F}_{\tau-1}$ and $\|\boldsymbol{\phi}_\tau\|_2 \leq 1$. Let $\mathbf{\Lambda}_k = I + \sum_{\tau=1}^k \boldsymbol{\phi}_\tau \boldsymbol{\phi}_\tau^\top$. Then for any $\delta > 0$, with probability at least $1 - \delta$, for all $k \geq 0$, and any $V \in \mathscr{F}$ so that $\sup_s |V(s)| \leq H$, we have:*

$$\left\| \mathbf{\Lambda}_k^{-1/2} \sum_{\tau=1}^k \boldsymbol{\phi}_\tau (V(s_\tau) - \mathbb{E}[V(s_\tau) | \mathcal{F}_{\tau-1}]) \right\|^2 \leq 4H^2 \left(\frac{d}{2} \log(1 + k) + \log \frac{|\mathcal{N}_\epsilon|}{\delta} \right) + 8k^2\epsilon^2$$

where $\mathcal{N}_\epsilon$ is the ϵ-covering of $\mathscr{F}$ with respect to dist_∞.

Lemma C.4 (Full version of Lemma 5.1)**.** *Let $\mathcal{E}_Q$ denote the event that, for all $h \in [H]$ and $V \in \mathscr{F}_{h+1}$ simultaneously,*

$$\left\| \mathbf{\Lambda}_h^{-1/2} \sum_{i=1}^{\iota_\epsilon} \sum_{\tau=1}^{K_i} \boldsymbol{\phi}_{h,\tau}^i \left[V(s_{h+1,\tau}^i) - \mathbb{E}_h[V](s_{h,\tau}^i, a_{h,\tau}^i) \right] \right\|_2 \leq cH \sqrt{d \log(1 + dHK_{\mathrm{tot}}) + \log H/\delta}$$

for a universal constant c and where $K_{\mathrm{tot}} = \sum_{i=1}^{\iota_\epsilon} K_i$ and $\mathscr{F}_{h+1} := \mathscr{F}(\mathbf{\Lambda}_{h+1}, 2H\sqrt{dK_{\mathrm{tot}}})$. Then $\mathbb{P}[\mathcal{E}_Q] \geq 1 - \delta$.

Proof. This is a consequence of Lemma C.2 and Lemma C.3. Fix some ϵ, then by Lemma C.2 we have

$$\mathsf{N}(\mathscr{F}_{h+1}, \mathrm{dist}_\infty, \epsilon) \leq d \log(1 + 2H\sqrt{dK_{\mathrm{tot}}}/\epsilon)$$

and by Lemma C.3, with probability $1 - \delta$, for all $V \in \mathscr{F}_{h+1}$ simultaneously,

$$\begin{aligned} \left\| \mathbf{\Lambda}_h^{-1/2} \sum_{i=1}^{\iota_\epsilon} \sum_{\tau=1}^{K_i} \boldsymbol{\phi}_{h,\tau}^i [V(s_{h+1,\tau}^i) - \mathbb{E}_h[V](s_{h,\tau}^i, a_{h,\tau}^i)] \right\|_2^2 &\leq 4H^2 \left(\frac{d}{2} \log(1 + K_{\mathrm{tot}}) + \log \frac{|\mathcal{N}_\epsilon|}{\delta} \right) + 8K_{\mathrm{tot}}^2 \epsilon^2 \\ &\leq 4H^2 \left(\frac{d}{2} \log(1 + K_{\mathrm{tot}}) + d \log(1 + \frac{2H\sqrt{dK_{\mathrm{tot}}}}{\epsilon}) + \log \frac{1}{\delta} \right) + 8K_{\mathrm{tot}}^2 \epsilon^2. \end{aligned}$$

Note also that, due to our data collection procedure collecting samples independently for each h, we have that $\mathbf{\Lambda}_h$ and $\{\{(\boldsymbol{\phi}_{h,\tau}^i, s_{h+1,\tau}^i)\}_{\tau=1}^{K_i}\}_{i=1}^{\iota_\epsilon}$ are uncorrelated with $\mathbf{\Lambda}_{h+1}$, so, unlike in (Jin et al., 2020b), no union bound over possible $\mathbf{\Lambda}_{h+1}$ is needed. Choosing $\epsilon = \sqrt{\frac{H^2 d}{8K_{\mathrm{tot}}^2}}$, we can bound this as

$$\leq cH^2 \left(d \log(1 + dHK_{\mathrm{tot}}) + \log \frac{1}{\delta} \right)$$

for a universal constant c. The result follows by a union bound over all h. □

C.2. Optimism

As noted above, throughout this section we will let $V^\star, V^\pi, Q^\star, Q^\pi$ denote value functions defined with respect to a generic reward function r, and V, Q the value function estimates maintained by RFLIN-PLAN when called with reward function r.

Lemma C.5. *On the event $\mathcal{E}_Q$, for all s, a, h and all r satisfying Definition 3.1,*

$$|\langle \boldsymbol{\phi}(s,a), \widehat{\boldsymbol{w}}_h\rangle - r_h(s,a) - \mathbb{E}_h[V_{h+1}](s,a)| \le \widetilde{\beta}\|\boldsymbol{\phi}(s,a)\|_{\boldsymbol{\Lambda}_h^{-1}}$$

where $\widetilde{\beta} = cH\sqrt{d\log(1+dHK_{\mathrm{tot}}) + \log H/\delta}$.

Proof. First, note that for any r satisfying Definition 3.1, by Lemma C.1, we will have that $V_h \in \mathscr{F}_h$, where $\mathscr{F}_h$ is defined as in Lemma C.4. Therefore, on $\mathcal{E}_Q$, we have for any r,

$$\left\|\boldsymbol{\Lambda}_h^{-1/2}\sum_{i=1}^{\iota_\epsilon}\sum_{\tau=1}^{K_i}\boldsymbol{\phi}_{h,\tau}^i\left[V_{h+1}(s_{h+1,\tau}^i) - \mathbb{E}_h[V_{h+1}](s_{h,\tau}^i, a_{h,\tau}^i)\right]\right\|_2 \le \widetilde{\beta}. \tag{C.1}$$

By definition:

$$\widehat{\boldsymbol{w}}_h = \boldsymbol{\Lambda}_h^{-1}\sum_{i=1}^{\iota_\epsilon}\sum_{\tau=1}^{K_i}\boldsymbol{\phi}_{h,\tau}^i(r(s_{h,\tau}^i, a_{h,\tau}^i) + V_{h+1}(s_{h+1,\tau}^i)).$$

Furthermore, we recall that $r_h(s_{h,\tau}^i, a_{h,\tau}^i) = \langle \boldsymbol{\theta}_h, \boldsymbol{\phi}_{h,\tau}^i\rangle$. Thus,

$$\begin{aligned}
\langle \boldsymbol{\phi}(s,a), \widehat{\boldsymbol{w}}_h\rangle &= \langle \boldsymbol{\phi}(s,a), \boldsymbol{\Lambda}_h^{-1}\sum_{i=1}^{\iota_\epsilon}\sum_{\tau=1}^{K_i}\boldsymbol{\phi}_{h,\tau}^i(r(s_{h,\tau}^i, a_{h,\tau}^i) + V_{h+1}(s_{h+1,\tau}^i))\rangle \\
&= \underbrace{\langle \boldsymbol{\phi}(s,a), \boldsymbol{\Lambda}_h^{-1}\sum_{i=1}^{\iota_\epsilon}\sum_{\tau=1}^{K_i}\boldsymbol{\phi}_{h,\tau}^i(\boldsymbol{\phi}_{h,\tau}^i)^\top\boldsymbol{\theta}_h\rangle}_{(a)} \\
&\quad + \underbrace{\langle \boldsymbol{\phi}(s,a), \boldsymbol{\Lambda}_h^{-1}\sum_{i=1}^{\iota_\epsilon}\sum_{\tau=1}^{K_i}\boldsymbol{\phi}_{h,\tau}^i(V_{h+1}(s_{h+1,\tau}^i) - \mathbb{E}[V_{h+1}](s_{h,\tau}^i, a_{h,\tau}^i))\rangle}_{(b)} \\
&\quad + \underbrace{\langle \boldsymbol{\phi}(s,a), \boldsymbol{\Lambda}_h^{-1}\sum_{i=1}^{\iota_\epsilon}\sum_{\tau=1}^{K_i}\boldsymbol{\phi}_{h,\tau}^i\mathbb{E}[V_{h+1}](s_{h,\tau}^i, a_{h,\tau}^i)\rangle}_{(c)}.
\end{aligned}$$

Now,

$$(a) = \langle \boldsymbol{\phi}(s,a), \boldsymbol{\theta}_h\rangle - \langle \boldsymbol{\phi}(s,a), \boldsymbol{\Lambda}_h^{-1}\boldsymbol{\theta}_h\rangle = r_h(s,a) - \langle \boldsymbol{\phi}(s,a), \boldsymbol{\Lambda}_h^{-1}\boldsymbol{\theta}_h\rangle$$

and, using the linear MDP assumption, Definition 3.1,

$$\begin{aligned}
(c) &= \langle \boldsymbol{\phi}(s,a), \boldsymbol{\Lambda}_h^{-1}\sum_{i=1}^{\iota_\epsilon}\sum_{\tau=1}^{K_i}\boldsymbol{\phi}_{h,\tau}^i(\boldsymbol{\phi}_{h,\tau}^i)^\top\int V_{h+1}(s')\mathrm{d}\boldsymbol{\mu}_h(s')\rangle \\
&= \langle \boldsymbol{\phi}(s,a), \int V_{h+1}(s')\mathrm{d}\boldsymbol{\mu}_h(s')\rangle - \langle \boldsymbol{\phi}(s,a), \boldsymbol{\Lambda}_h^{-1}\int V_{h+1}(s')\mathrm{d}\boldsymbol{\mu}_h(s')\rangle \\
&= \mathbb{E}_h[V_{h+1}](s,a) - \langle \boldsymbol{\phi}(s,a), \boldsymbol{\Lambda}_h^{-1}\int V_{h+1}(s')\mathrm{d}\boldsymbol{\mu}_h(s')\rangle.
\end{aligned}$$

Thus,

$$|\langle \boldsymbol{\phi}(s,a), \widehat{\boldsymbol{w}}_h\rangle - r_h(s,a) - \mathbb{E}_h[V_{h+1}](s,a)| \le (b) + \left|\langle \boldsymbol{\phi}(s,a), \boldsymbol{\Lambda}_h^{-1}\boldsymbol{\theta}_h\rangle\right|$$

$$+ \left| \langle \boldsymbol{\phi}(s,a), \boldsymbol{\Lambda}_h^{-1} \int V_{h+1}(s') \mathrm{d}\boldsymbol{\mu}_h(s') \rangle \right|.$$

By Cauchy-Schwartz and (C.1):

$$(b) \le \|\boldsymbol{\phi}(s,a)\|_{\boldsymbol{\Lambda}_h^{-1}} \left\| \boldsymbol{\Lambda}_h^{-1/2} \sum_{i=1}^{\iota_\epsilon} \sum_{\tau=1}^{K_i} \boldsymbol{\phi}_{h,\tau}^i (V_{h+1}(s_{h+1,\tau}^i) - \mathbb{E}_h[V_{h+1}](s_{h,\tau}^i, a_{h,\tau}^i)) \right\|_2 \le \widetilde{\beta} \|\boldsymbol{\phi}(s,a)\|_{\boldsymbol{\Lambda}_h^{-1}}.$$

Furthermore, we can bound

$$\left| \langle \boldsymbol{\phi}(s,a), \boldsymbol{\Lambda}_h^{-1} \boldsymbol{\theta}_h \rangle \right| \le \sqrt{d} \|\boldsymbol{\phi}(s,a)\|_{\boldsymbol{\Lambda}_h^{-1}}$$

and

$$\begin{aligned} \left| \langle \boldsymbol{\phi}(s,a), \boldsymbol{\Lambda}_h^{-1} \int V_{h+1}(s') \mathrm{d}\boldsymbol{\mu}_h(s') \rangle \right| &\le \|\boldsymbol{\phi}(s,a)\|_{\boldsymbol{\Lambda}_h^{-1}} \|\boldsymbol{\Lambda}_h^{-1/2}\|_{\mathrm{op}} \left\| \int V_{h+1}(s') \mathrm{d}\boldsymbol{\mu}_h(s') \right\|_2 \\ &\le H\sqrt{d} \|\boldsymbol{\phi}(s,a)\|_{\boldsymbol{\Lambda}_h^{-1}} \end{aligned}$$

where the final inequality uses the linear MDP assumption, Definition 3.1, and that $\boldsymbol{\Lambda}_h \succeq I$. The result follows by combining these bounds. □

Lemma C.6. *On the event $\mathcal{E}_Q$ and assuming that $\beta \ge \widetilde{\beta}$, it holds that $Q_h(s,a) \ge Q_h^\star(s,a)$ for all s, a, h.*

Proof. We will prove this by induction, starting at step H. Since the value function at $H+1$ is 0 by definition, $Q_H^\star(s,a) = r_H(s,a)$. Thus, Lemma C.5 gives

$$|\langle \boldsymbol{\phi}(s,a), \widehat{\boldsymbol{w}}_H \rangle - Q_H^\star(s,a)| \le \widetilde{\beta} \|\boldsymbol{\phi}(s,a)\|_{\boldsymbol{\Lambda}_H^{-1}}.$$

It follows that, since $\beta \ge \widetilde{\beta}$,

$$Q_H^\star(s,a) \le \min\{\langle \boldsymbol{\phi}(s,a), \widehat{\boldsymbol{w}}_H \rangle + \beta \|\boldsymbol{\phi}(s,a)\|_{\boldsymbol{\Lambda}_H^{-1}}, H\} = Q_H(s,a).$$

Now assume that $Q_{h+1}(s,a) \ge Q_{h+1}^\star(s,a)$ for all s, a. By the Bellman equation,

$$Q_h^\star(s,a) = r_h(s,a) + \mathbb{E}_h[V_{h+1}^\star](s,a).$$

Thus, Lemma C.5 gives

$$|\langle \boldsymbol{\phi}(s,a), \widehat{\boldsymbol{w}}_h \rangle - Q_h^\star(s,a) - \mathbb{E}_h[V_{h+1} - V_{h+1}^\star](s,a)| \le \beta \|\boldsymbol{\phi}(s,a)\|_{\boldsymbol{\Lambda}_h^{-1}}.$$

By the inductive assumption, $\mathbb{E}_h[V_{h+1} - V_{h+1}^\star](s,a) \ge 0$, so we can rearrange this to get

$$Q_h^\star(s,a) \le \min\{\langle \boldsymbol{\phi}(s,a), \widehat{\boldsymbol{w}}_h \rangle + \widetilde{\beta} \|\boldsymbol{\phi}(s,a)\|_{\boldsymbol{\Lambda}_h^{-1}}, H\} = Q_h(s,a).$$

□

D. Lower Bound for PAC Reinforcement Learning

Theorem 5 (Lower Bound on Adaptive Linear Regression)**.** *Let $\Phi = \mathcal{S}^{d-1}$ and $\Theta = \{-\mu, \mu\}^d$ for some $\mu \in (0, \frac{1}{20\sqrt{d}}]$. Consider the query model where at every step $t = 1, \dots, T$ we choose some $\boldsymbol{\phi}_t \in \Phi$, and observe*

$$y_t \sim \mathrm{Bernoulli}(1/2 + \langle \boldsymbol{\theta}, \boldsymbol{\phi}_t \rangle) \tag{D.1}$$

for $\boldsymbol{\theta} \in \Theta$. Then

$$\inf_{\widehat{\boldsymbol{\theta}}, \pi} \max_{\boldsymbol{\theta} \in \Theta} \mathbb{E}_{\boldsymbol{\theta}}[\|\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}\|_2^2] \ge \frac{d\mu^2}{2} \left(1 - \sqrt{\frac{20T\mu^2}{d}}\right).$$

where the infimum is over all measurable estimators $\widehat{\boldsymbol{\theta}}$ and measurable (but possibly adaptive) query rules π, and $\mathbb{E}_{\boldsymbol{\theta}}[\cdot]$ denotes the expectation over the randomness in the observations and decision rules if $\boldsymbol{\theta}$ is the true instance. In particular, if $T \geq d^2$, choosing $\mu = \sqrt{d/700T}$, we get

$$\inf_{\widehat{\boldsymbol{\theta}},\pi} \max_{\boldsymbol{\theta}\in\Theta} \mathbb{E}_{\boldsymbol{\theta}}[\|\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}\|_2^2] \geq 0.00059 \cdot d^2/T.$$

Proof. The proof of this result follows closely the proof of Theorem 3 of (Shamir, 2013). Clearly,

$$\begin{aligned}
\max_{\boldsymbol{\theta}\in\Theta} \mathbb{E}_{\boldsymbol{\theta}}[\|\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}\|_2^2] &\geq \mathbb{E}_{\boldsymbol{\theta}\sim\mathrm{unif}(\Theta)}\mathbb{E}_{\boldsymbol{\theta}}[\|\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}\|_2^2] \\
&= \mathbb{E}_{\boldsymbol{\theta}\sim\mathrm{unif}(\Theta)}\mathbb{E}_{\boldsymbol{\theta}}\left[\sum_{i=1}^d (\boldsymbol{\theta}_i - \widehat{\boldsymbol{\theta}}_i)^2\right] \\
&\geq \mathbb{E}_{\boldsymbol{\theta}\sim\mathrm{unif}(\Theta)}\mathbb{E}_{\boldsymbol{\theta}}\left[\mu^2 \sum_{i=1}^d \mathbb{I}\{\boldsymbol{\theta}_i\widehat{\boldsymbol{\theta}}_i < 0\}\right].
\end{aligned}$$

As in (Shamir, 2013), we assume that the query strategy is deterministic conditioned on the past: $\boldsymbol{\phi}_t$ is a deterministic function of $y_1, \ldots, y_{t-1}, \boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_{t-1}$, which is without loss of generality. We then need the following result.

Lemma D.1 (Lemma 4 of (Shamir, 2013))**.** *Let $\boldsymbol{\theta}$ be a random vector, none of whose coordinates is supported on 0, and let $y_1, y_2, \ldots, y_T$ be a sequence of query values obtained by a deterministic strategy returning a point $\widehat{\boldsymbol{\theta}}$ (that is, $\boldsymbol{\phi}_t$ is a deterministic function of $y_1, \ldots, y_{t-1}, \boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_{t-1}$, and $\widehat{\boldsymbol{\theta}}$ is a deterministic function of $y_1, \ldots, y_T$). Then we have*

$$\mathbb{E}_{\boldsymbol{\theta}\sim\mathrm{unif}(\Theta)}\mathbb{E}_{\boldsymbol{\theta}}\left[\sum_{i=1}^d \mathbb{I}\{\boldsymbol{\theta}_i\widehat{\boldsymbol{\theta}}_i < 0\}\right] \geq \frac{d}{2}\left(1 - \sqrt{\frac{1}{d}\sum_{i=1}^d\sum_{t=1}^T U_{t,i}}\right)$$

where

$$U_{t,i} = \sup_{\boldsymbol{\theta}_j, j\neq i} \mathrm{KL}\Big(\mathbb{P}(y_t|\boldsymbol{\theta}_i > 0, \{\boldsymbol{\theta}_j\}_{j\neq i}, \{y_s\}_{s=1}^{t-1}) || \mathbb{P}(y_t|\boldsymbol{\theta}_i < 0, \{\boldsymbol{\theta}_j\}_{j\neq i}, \{y_s\}_{s=1}^{t-1})\Big).$$

In our setting, y_t is distributed as in (D.1). Thus, we have that

$$U_{t,i} = \sup_{\boldsymbol{\theta}_j, j\neq i} \mathrm{KL}\Big(\mathrm{Bernoulli}(1/2 + \sum_{j\neq i}\boldsymbol{\theta}_j\boldsymbol{\phi}_{t,j} + \mu\boldsymbol{\phi}_{t,i}) || \mathrm{Bernoulli}(1/2 + \sum_{j\neq i}\boldsymbol{\theta}_j\boldsymbol{\phi}_{t,j} - \mu\boldsymbol{\phi}_{t,i})\Big).$$

Note that:

Lemma D.2 (Lemma 2.7 of (Tsybakov, 2009))**.**

$$\mathrm{KL}(\mathrm{Bernoulli}(p)||\mathrm{Bernoulli}(q)) \leq \frac{(p-q)^2}{q(1-q)}.$$

Thus, by Lemma D.2, we can bound

$$U_{t,i} \leq \frac{(2\mu\boldsymbol{\phi}_{t,i})^2}{(1/2 + \sum_{j\neq i}\boldsymbol{\theta}_j\boldsymbol{\phi}_{t,j} - \mu\boldsymbol{\phi}_{t,i})(1 - 1/2 - \sum_{j\neq i}\boldsymbol{\theta}_j\boldsymbol{\phi}_{t,j} + \mu\boldsymbol{\phi}_{t,i})} \leq 20\mu^2\boldsymbol{\phi}_{t,i}^2$$

where the last inequality follows by our assumption that $\mu \leq \frac{1}{20\sqrt{d}}$ and since $\boldsymbol{\phi}_t \in \mathcal{S}^{d-1}$, which implies that $1/2 + \sum_{j\neq i}\boldsymbol{\theta}_j\boldsymbol{\phi}_{t,j} - \mu\boldsymbol{\phi}_{t,i} \geq 9/20$ and $1 - 1/2 - \sum_{j\neq i}\boldsymbol{\theta}_j\boldsymbol{\phi}_{t,j} + \mu\boldsymbol{\phi}_{t,i} \geq 9/20$.

By Lemma D.1 and this calculation, we can lower bound

$$\mathbb{E}_{\boldsymbol{\theta}\sim\mathrm{unif}(\Theta)}\mathbb{E}_{\boldsymbol{\theta}}\left[\mu^2\sum_{i=1}^d \mathbb{I}\{\boldsymbol{\theta}_i\widehat{\boldsymbol{\theta}}_i < 0\}\right] \geq \frac{d\mu^2}{2}\left(1 - \sqrt{\frac{1}{d}\sum_{i=1}^d\sum_{t=1}^T U_{t,i}}\right)$$

$$\geq \frac{d\mu^2}{2}\left(1 - \sqrt{\frac{1}{d}\sum_{i=1}^{d}\sum_{t=1}^{T} 20\mu^2\phi_{t,i}^2}\right)$$
$$= \frac{d\mu^2}{2}\left(1 - \sqrt{\frac{20T\mu^2}{d}}\right)$$

where the final equality follows since $\phi_t \in \mathcal{S}^{d-1}$. This proves the first conclusion. The second conclusion follows by our choice of μ. □

Proof of Theorem 2. We consider the setting of instances in Theorem 5 with $\Phi = \mathcal{S}^{d-1}$ and $\Theta = \{-\mu, \mu\}^d$. We first show how estimation error relates to finding ϵ-good arms. Let $\phi^\star(\boldsymbol{\theta})$ denote the optimal arm for $\boldsymbol{\theta}$. Clearly, $\phi^\star(\boldsymbol{\theta}) = \boldsymbol{\theta}/\|\boldsymbol{\theta}\|_2 = \boldsymbol{\theta}/(\sqrt{d}\mu)$ and $\phi^\star(\boldsymbol{\theta})^\top\boldsymbol{\theta} = \sqrt{d}\mu$. Now consider a distribution $\pi \in \triangle_\Phi$. By definition, we have that π is ϵ-optimal if

$$\mathbb{E}_{\boldsymbol{\phi}\sim\pi}[\boldsymbol{\theta}^\top\boldsymbol{\phi}] = \boldsymbol{\theta}^\top\mathbb{E}_{\boldsymbol{\phi}\sim\pi}[\boldsymbol{\phi}] \geq \sqrt{d}\mu - \epsilon.$$

For a policy π, let $\boldsymbol{\phi}_\pi := \mathbb{E}_{\boldsymbol{\phi}\sim\pi}[\boldsymbol{\phi}]$. Note that by the convexity of norms and Jensen's inequality:

$$\|\boldsymbol{\phi}_\pi\|_2^2 \leq \mathbb{E}_{\boldsymbol{\phi}\sim\pi}[\|\boldsymbol{\phi}\|_2^2] = 1.$$

Write $\boldsymbol{\phi}_\pi = \phi^\star(\boldsymbol{\theta}) + \Delta$. Then,

$$1 \geq \|\phi^\star(\boldsymbol{\theta}) + \Delta\|_2^2 = 1 + \|\Delta\|_2^2 + 2\phi^\star(\boldsymbol{\theta})^\top\Delta \implies \phi^\star(\boldsymbol{\theta})^\top\Delta \leq -\frac{1}{2}\|\Delta\|_2^2 \implies \boldsymbol{\theta}^\top\Delta \leq -\frac{\sqrt{d}\mu}{2}\|\Delta\|_2^2.$$

However, $\boldsymbol{\phi}_\pi^\top\boldsymbol{\theta} = \phi^\star(\boldsymbol{\theta})^\top\boldsymbol{\theta} + \Delta^\top\boldsymbol{\theta}$, so if π is ϵ-optimal, we have $\Delta^\top\boldsymbol{\theta} \geq -\epsilon$ which implies

$$-\frac{\sqrt{d}\mu}{2}\|\Delta\|_2^2 \geq -\epsilon.$$

In other words, if π is ϵ-optimal for $\boldsymbol{\theta}$, then $\boldsymbol{\phi}_\pi = \phi^\star(\boldsymbol{\theta}) + \Delta$ for some Δ with $\|\Delta\|_2^2 \leq \frac{2\epsilon}{\sqrt{d}\mu}$, so

$$\boldsymbol{\phi}_\pi = \phi^\star(\boldsymbol{\theta}) + \Delta = \frac{\boldsymbol{\theta}}{\sqrt{d}\mu} + \Delta \implies \boldsymbol{\theta} = \sqrt{d}\mu(\boldsymbol{\phi}_\pi - \Delta).$$

Now assume that we have a $\widehat{\pi}$ which is ϵ-optimal, and denote $\widehat{\boldsymbol{\phi}} := \boldsymbol{\phi}_{\widehat{\pi}}$. Let $\boldsymbol{\theta} = \sqrt{d}\mu(\widehat{\boldsymbol{\phi}} - \Delta)$ as above. Define the following estimator

$$\widehat{\boldsymbol{\theta}} = \begin{cases} \boldsymbol{\theta}' & \text{if } \exists\boldsymbol{\theta}' \in \Theta \text{ with } \boldsymbol{\theta}' = \sqrt{d}\mu(\widehat{\boldsymbol{\phi}} - \Delta') \text{ for some } \Delta' \in \mathbb{R}^d, \|\Delta'\|_2^2 \leq \frac{2\epsilon}{\sqrt{d}\mu} \\ \text{any } \boldsymbol{\theta}' \in \Theta & \text{otherwise} \end{cases}$$

If $\widehat{\boldsymbol{\phi}}$ is actually ϵ-optimal for some $\boldsymbol{\theta} \in \Theta$, then the first condition is met and

$$\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}\|_2 = \|\sqrt{d}\mu(\widehat{\boldsymbol{\phi}} - \Delta') - \sqrt{d}\mu(\widehat{\boldsymbol{\phi}} - \Delta)\|_2 \leq 2\sqrt{d}\mu\sqrt{\frac{2\epsilon}{\sqrt{d}\mu}} = \sqrt{8\sqrt{d}\mu\epsilon}.$$

Thus, if we can find an ϵ-good arm for $\boldsymbol{\theta}$, we can estimate $\boldsymbol{\theta}$ up to tolerance $\sqrt{8\sqrt{d}\mu\epsilon}$.

Now set $\mu = \sqrt{d/700K}$. Let $\widehat{\boldsymbol{\theta}}$ denote the estimator constructed from $\widehat{\boldsymbol{\phi}}$ as outlined above. Let $\mathcal{E}$ be the event that $\widehat{\boldsymbol{\phi}}$ is ϵ-good for $\boldsymbol{\theta}$ and note that regardless of whether or not we are on $\mathcal{E}$, $\|\widehat{\boldsymbol{\theta}}\|_2 \leq d/\sqrt{700K}$. Thus,

$$\mathbb{E}_{\boldsymbol{\theta}}[\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}\|_2^2] = \mathbb{E}_{\boldsymbol{\theta}}[\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}\|_2^2 \cdot \mathbb{I}\{\mathcal{E}\} + \|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}\|_2^2 \cdot \mathbb{I}\{\mathcal{E}^c\}]$$
$$\leq \frac{8d\epsilon}{\sqrt{700K}} + \frac{4d^2}{700K}\mathbb{P}_{\boldsymbol{\theta}}[\mathcal{E}^c]$$

where the first inequality follows by our observation above that any ϵ-good $\widehat{\phi}$ yields an estimate of $\boldsymbol{\theta}$ up to tolerance $\sqrt{2d\epsilon/\sqrt{K}}$.

However, by what we have shown above, there exists some $\boldsymbol{\theta}$ such that if we collect (no more than) K samples,

$$\mathbb{E}_{\boldsymbol{\theta}}[\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}\|_2^2] \geq 0.00059 \cdot d^2/K.$$

This is a contradiction unless

$$\frac{8d\epsilon}{\sqrt{700K}} + \frac{4d^2}{700K}\mathbb{P}_{\boldsymbol{\theta}}[\mathcal{E}^c] \geq 0.00059\frac{d^2}{K} \iff \mathbb{P}_{\boldsymbol{\theta}}[\mathcal{E}^c] \geq 0.10325 - \frac{2\sqrt{700}\epsilon\sqrt{K}}{d}.$$

It follows that if

$$0.10325 - \frac{2\sqrt{700}\epsilon\sqrt{K}}{d} \geq 0.1 \iff (\frac{0.00325}{2\sqrt{700}})^2 \cdot \frac{d^2}{\epsilon^2} \geq K$$

we have that $\mathbb{P}_{\boldsymbol{\theta}}[\mathcal{E}^c] \geq 0.1$. □

D.1. Mapping to Linear MDPs

We next show that the linear bandit instance of Theorem 2 with parameter $\boldsymbol{\theta}$ (for $\boldsymbol{\theta} \in \Theta$ as in Theorem 2) can be mapped to a linear MDP with state space $\mathcal{S} = \{s_0, s_1, \bar{s}_2, \ldots, \bar{s}_{d+1}\}$, action space $\mathcal{A} = \mathcal{S}^{d-1} \cup \{\boldsymbol{e}_{d+1}/2\}$, parameters

$$\boldsymbol{\theta}_1 = \mathbf{0}, \quad \boldsymbol{\theta}_h = \boldsymbol{e}_1, h \geq 2$$
$$\boldsymbol{\mu}_1(s_1) = [2\boldsymbol{\theta}, 1], \quad \boldsymbol{\mu}_1(\bar{s}_i) = \frac{1}{d}[-2\boldsymbol{\theta}, 1], \quad \boldsymbol{\mu}_h(s_i) = \boldsymbol{e}_i, h \geq 2$$

and feature vectors

$$\boldsymbol{\phi}(s_0, \boldsymbol{e}_{d+1}/2) = \boldsymbol{e}_{d+1}/2, \quad \boldsymbol{\phi}(s_0, \widetilde{\boldsymbol{\phi}}) = [\widetilde{\boldsymbol{\phi}}/2, 1/2], \quad \forall \widetilde{\boldsymbol{\phi}} \in \mathcal{S}^{d-1}$$
$$\boldsymbol{\phi}(s_1, \widetilde{\boldsymbol{\phi}}) = \boldsymbol{e}_1, \quad \boldsymbol{\phi}(\bar{s}_i, \widetilde{\boldsymbol{\phi}}) = \boldsymbol{e}_i, i \geq 2, \quad \forall \widetilde{\boldsymbol{\phi}} \in \mathcal{A}.$$

Note that, if we take action $\widetilde{\boldsymbol{\phi}}$ in state s_0, our expected episode reward is

$$P_1(s_1|s_0, \widetilde{\boldsymbol{\phi}}) \cdot H + \sum_{i=2}^{d+1} P_1(\bar{s}_i|s_0, \widetilde{\boldsymbol{\phi}}) \cdot 0 = H(\langle \boldsymbol{\theta}, \widetilde{\boldsymbol{\phi}} \rangle + 1/2)$$

since we always acquire a reward of 1 in any state s_1, and a reward of 0 in any state $\bar{s}_i$, and the reward distribution is Bernoulli.

Lemma D.3. *The MDP constructed above is a valid linear MDP as defined in Definition 3.1.*

Proof. For $\widetilde{\boldsymbol{\phi}} \in \mathcal{S}^{d-1}$ we have,

$$P_1(s_1|s_0, \widetilde{\boldsymbol{\phi}}) = \langle \boldsymbol{\phi}(s_0, \widetilde{\boldsymbol{\phi}}), \boldsymbol{\mu}_1(s_1) \rangle = \langle \boldsymbol{\theta}, \widetilde{\boldsymbol{\phi}} \rangle + 1/2 \geq 0$$
$$P_1(\bar{s}_i|s_0, \widetilde{\boldsymbol{\phi}}) = \langle \boldsymbol{\phi}(s_0, \widetilde{\boldsymbol{\phi}}), \boldsymbol{\mu}_1(\bar{s}_i) \rangle = \frac{1}{d}(-\langle \boldsymbol{\theta}, \widetilde{\boldsymbol{\phi}} \rangle + 1/2) \geq 0$$

where the inequality follows since $|\langle \boldsymbol{\theta}, \widetilde{\boldsymbol{\phi}} \rangle| \leq \mathcal{O}(1/d)$ for all $\widetilde{\boldsymbol{\phi}} \in \mathcal{S}^{d-1}$, by the choice of $\boldsymbol{\theta}$ in Theorem 2, and our condition that $K \geq d^2$. In addition,

$$P_1(s_1|s_0, \widetilde{\boldsymbol{\phi}}) + \sum_{i=2}^{d+1} P_1(\bar{s}_i|s_0, \widetilde{\boldsymbol{\phi}}) = \langle \boldsymbol{\theta}_\star, \widetilde{\boldsymbol{\phi}} \rangle + 1/2 + d \cdot \frac{1}{d}(-\langle \boldsymbol{\theta}_\star, \widetilde{\boldsymbol{\phi}} \rangle + 1/2) = 1.$$

Thus, $P_1(\cdot|s_0, \widetilde{\boldsymbol{\phi}})$ is a valid probability distribution for $\widetilde{\boldsymbol{\phi}} \in \mathcal{S}^{d-1}$. A similar calculation shows the same for action $\boldsymbol{e}_{d+1}/2$. It is obvious that $P_h(\cdot|s, \widetilde{\boldsymbol{\phi}})$ is a valid distribution for all s and $\widetilde{\boldsymbol{\phi}} \in \mathcal{A}$.

It remains to check the normalization bounds. Clearly, by our construction of the feature vectors, $\|\boldsymbol{\phi}(s,a)\|_2 \leq 1$ for all s and a. It is also obvious that $\|\boldsymbol{\theta}_1\|_2 \leq \sqrt{d}$ and $\|\boldsymbol{\theta}_h\|_2 \leq \sqrt{d}$. Finally,

$$\||\boldsymbol{\mu}_1(\mathcal{S})|\|_2 = \left\| \sum_{s \in \mathcal{S} \setminus s_0} |\boldsymbol{\mu}_1(s)| \right\|_2 = \|[2\boldsymbol{\theta}, 1] + d \cdot \frac{1}{d}[2\boldsymbol{\theta}, 1]\|_2 \leq \sqrt{d}.$$

Thus, all normalization bounds are met, so this is a valid linear MDP. □

Lower bounding the performance of low-regret algorithms. Assume that we have access to the linear bandit instance constructed in Theorem 2. That is, at every timestep t we can choose an arm $\widetilde{\boldsymbol{\phi}}_t \in \mathcal{S}^{d-1}$ and obtain and observe reward $y_t \sim \text{Bernoulli}(\langle \boldsymbol{\theta}, \widetilde{\boldsymbol{\phi}}_t \rangle + 1/2)$. Using the mapping above, we can use this bandit to simulate a linear MDP as follows:

1. Start in state s_0 and choose any action $\widetilde{\boldsymbol{\phi}}_t \in \mathcal{A}$

2. Play action $\widetilde{\boldsymbol{\phi}}_t$ in our linear bandit. If reward obtained is $y_t = 1$, then in the MDP transition to any of the states s_1. If the reward obtained is $y_t = 0$ transition to any of the states $\bar{s}_2, \ldots, \bar{s}_{d+1}$, each with probability $1/d$. If the chosen action was $\widetilde{\boldsymbol{\phi}}_t = \boldsymbol{e}_{d+1}/2$, then play any action in the linear bandit and transition to state s_1 with probability 1/2 and $\bar{s}_2, \ldots, \bar{s}_{d+1}$ with probability $1/2d$, regardless of y_t

3. For the next $H - 1$ steps, take any action in the state in which you end up, transition back to the same state with probability 1, and receive reward of 1 if you are in s_1, and reward of 0 if you are in $\bar{s}_2, \ldots, \bar{s}_{d+1}$.

Note that this MDP has precisely the transition and reward structure as the MDP constructed above.

Lemma D.4. *Assume π is ϵ-optimal in the MDP constructed above. Then $\pi_1(\cdot|s_0)$ is ϵ/H-optimal on the linear bandit instance with parameter $\boldsymbol{\theta}$ and action set $\mathcal{S}^{d-1}$.*

Proof. Note that the value of π in the linear MDP is given by

$$V_0^\pi = H \cdot \left(\sum_{\widetilde{\boldsymbol{\phi}} \in \mathcal{S}^{d-1}} \pi_1(\widetilde{\boldsymbol{\phi}}|s_0)(\langle \widetilde{\boldsymbol{\phi}}, \boldsymbol{\theta} \rangle + 1/2) + \pi_1(\boldsymbol{e}_{d+1}/2|s_0)/2 \right) = H \cdot \left(\sum_{\widetilde{\boldsymbol{\phi}} \in \mathcal{S}^{d-1}} \pi_1(\widetilde{\boldsymbol{\phi}}|s_0)\langle \widetilde{\boldsymbol{\phi}}, \boldsymbol{\theta} \rangle + 1/2 \right)$$

and the optimal policy is $\pi_1(a^\star|s_0) = 1$, for $a^\star = \arg\max_{\widetilde{\boldsymbol{\phi}} \in \mathcal{S}^{d-1}} \langle \widetilde{\boldsymbol{\phi}}, \boldsymbol{\theta} \rangle$, and has value $V_0^\star = \langle a^\star, \boldsymbol{\theta}_\star \rangle + 1/2$. It follows that if π is ϵ-optimal, then

$$\begin{aligned} H \cdot \left(\sum_{\widetilde{\boldsymbol{\phi}} \in \mathcal{S}^{d-1}} \pi_1(\widetilde{\boldsymbol{\phi}}|s_0)\langle \widetilde{\boldsymbol{\phi}}, \boldsymbol{\theta} \rangle + 1/2 \right) &\geq H \cdot (\langle a^\star, \boldsymbol{\theta} \rangle + 1/2) - \epsilon \\ \iff \sum_{\widetilde{\boldsymbol{\phi}} \in \mathcal{S}^{d-1}} \pi_1(\widetilde{\boldsymbol{\phi}}|s_0)\langle \widetilde{\boldsymbol{\phi}}, \boldsymbol{\theta} \rangle + 1/2 &\geq \langle a^\star, \boldsymbol{\theta} \rangle + 1/2 - \epsilon/H \end{aligned}$$

This implies that $\pi_1(\cdot|s_0)$ is ϵ/H-optimal for the linear bandit with parameter $\boldsymbol{\theta}$ and action set $\mathcal{S}^{d-1}$ (note that any mass $\pi(\cdot|s_0)$ places on arm $\boldsymbol{e}_{d+1}/2$ can be instead allocated to any arm in $\mathcal{S}^{d-1}$ without changing this result). □

Proof of Corollary 1. Consider running the above procedure for some number of steps. By Lemma D.4, if we can identify an ϵ-optimal policy in this MDP, we can use it to determine an ϵ/H-optimal policy on our bandit instance. As we have used no extra information other than samples from the linear bandit to construct this, it follows that to find an ϵ/H-optimal policy in the MDP, we must take at least the number of samples prescribed by Theorem 2 for $\epsilon \leftarrow \epsilon/H$. □

Algorithm 5 Collect Well-Conditioned Covariates

input: confidence δ, target minimum reachability ϵ, step h, tolerance γ^2
Set $m \leftarrow \log(2/\epsilon)$ and $\lambda \leftarrow \min\{1, \frac{\epsilon}{4m\gamma^2}\}$
// Use $\mathbf{\Lambda}_{h,k-1} = \lambda I + \sum_{\tau=1}^{k-1} \boldsymbol{\phi}_{h,\tau}\boldsymbol{\phi}_{h,\tau}^\top$ in Egs
$\{(\mathcal{X}_i, \mathcal{D}_i, \mathbf{\Lambda}_i)\}_{i=1}^m \leftarrow$ CoverTraj$(h, \delta, \gamma^2, m,$ Force$)$
return $\mathcal{D}_{\text{exp}} = \bigcup_{i=1}^m \mathcal{D}_i$

E. Well-Conditioned Covariates

Remark E.1 (Handling unknown ϵ)**.** Note that Algorithm 5 requires knowledge of ϵ, a lower bound on the achievable minimum eigenvalue. In general a tight lower bound may not be known. In such situations, one can repeatedly run Algorithm 5 starting with $\epsilon = 1/d$ and halving ϵ each time until covariates satisfying the desired lower bound are obtained. Once ϵ reaches an achievable level, Theorem 4 will apply and the desired covariates will be collected. Note that this procedure only adds a complexity of $\mathcal{O}(\log(1/\epsilon_0))$ to the complexity, where $\epsilon_0 = \sup_\pi \lambda_{\min}(\mathbb{E}_\pi[\boldsymbol{\phi}(s_h, a_h)\boldsymbol{\phi}(s_h, a_h)^\top])$.

Proof of Theorem 4. **Step 1: Collected Data is Full-Rank.** We first argue that $\{\mathcal{X}_i\}_{i=1}^m$ is well-spread on full d dimension. For the following, let $\mathcal{Y} = \cup_{i=1}^m \mathcal{X}_i$. We have, for any $\boldsymbol{v} \in \mathcal{S}^{d-1}$,

$$\begin{aligned}
&\sup_\pi \boldsymbol{v}^\top \left(\mathbb{E}_\pi[\boldsymbol{\phi}(s_h, a_h)\boldsymbol{\phi}(s_h, a_h)^\top]\right)\boldsymbol{v} \\
&\quad\le \sup_\pi \boldsymbol{v}^\top \left(\mathbb{E}_\pi[\boldsymbol{\phi}(s_h, a_h)\boldsymbol{\phi}(s_h, a_h)^\top \cdot \mathbb{I}\{\boldsymbol{\phi}(s_h, a_h) \in \mathcal{Y}\}]\right)\boldsymbol{v} \\
&\qquad + \sup_\pi \boldsymbol{v}^\top \left(\mathbb{E}_\pi[\boldsymbol{\phi}(s_h, a_h)\boldsymbol{\phi}(s_h, a_h)^\top \cdot \mathbb{I}\{\boldsymbol{\phi}(s_h, a_h) \in \mathcal{X}_{m+1}\}]\right)\boldsymbol{v} \\
&\quad\le \sup_\pi \boldsymbol{v}^\top \left(\mathbb{E}_\pi[\boldsymbol{\phi}(s_h, a_h)\boldsymbol{\phi}(s_h, a_h)^\top \cdot \mathbb{I}\{\boldsymbol{\phi}(s_h, a_h) \in \mathcal{Y}\}]\right)\boldsymbol{v} + \epsilon/2,
\end{aligned}$$

where the last inequality comes from the definition of $\mathcal{X}_{m+1}$, Theorem 3, and our choice of m. On the other hand, by assumption we have

$$\sup_\pi \boldsymbol{v}^\top \left(\mathbb{E}_\pi[\boldsymbol{\phi}(s_h, a_h)\boldsymbol{\phi}(s_h, a_h)^\top]\right)\boldsymbol{v} \ge \sup_\pi \lambda_{\min}\left(\mathbb{E}_\pi[\boldsymbol{\phi}(s_h, a_h)\boldsymbol{\phi}(s_h, a_h)^\top]\right) \ge \epsilon$$

Therefore, we must have

$$\max_{\boldsymbol{\phi}\in\mathcal{Y}} \|\boldsymbol{\phi}\|^2_{\boldsymbol{v}\boldsymbol{v}^\top} \ge \sup_\pi \boldsymbol{v}^\top \left(\mathbb{E}_\pi[\boldsymbol{\phi}(s_h, a_h)\boldsymbol{\phi}(s_h, a_h)^\top \cdot \mathbb{I}\{\boldsymbol{\phi}(s_h, a_h) \in \mathcal{Y}\}]\right)\boldsymbol{v} \ge \epsilon/2$$

Step 2: Lower Bounding Eigenvalue. The next step is to show the lower bound of the eigenvalue of $\sum_{i=1}^m \mathbf{\Lambda}_i$, which is equivalent to upper bounding the eigenvalue of $\left(\sum_{i=1}^m \mathbf{\Lambda}_i\right)^{-1}$.

Denote the eigenvectors of $\sum_{i=1}^m \mathbf{\Lambda}_i$ as $\{\boldsymbol{u}_j\}_{j=1}^d$. Now for any $\boldsymbol{u}_i$ and any corresponding $\boldsymbol{\phi} = \arg\max_{\boldsymbol{\phi}'\in\mathcal{Y}} \|\boldsymbol{\phi}'\|^2_{\boldsymbol{u}_j\boldsymbol{u}_j^\top}$, we have

$$\begin{aligned}
\boldsymbol{u}_j^\top \left(\sum_{i=1}^m \mathbf{\Lambda}_i\right)^{-1} \boldsymbol{u}_j &= \frac{1}{\|\boldsymbol{\phi}\|^2_{\boldsymbol{u}_j\boldsymbol{u}_j^\top}} (\boldsymbol{u}_j^\top\boldsymbol{\phi})\boldsymbol{u}_j^\top \left(\sum_{i=1}^m \mathbf{\Lambda}_i\right)^{-1} (\boldsymbol{u}_j^\top\boldsymbol{\phi})\boldsymbol{u}_i \\
&\le \frac{1}{\|\boldsymbol{\phi}\|^2_{\boldsymbol{u}_j\boldsymbol{u}_j^\top}} \boldsymbol{\phi}^\top \left(\sum_{i=1}^m \mathbf{\Lambda}_i\right)^{-1} \boldsymbol{\phi} \\
&\le 2\epsilon^{-1}\boldsymbol{\phi}^\top \left(\sum_{i=1}^m \mathbf{\Lambda}_i\right)^{-1} \boldsymbol{\phi} \\
&\le 2\epsilon^{-1}\boldsymbol{\phi}^\top \left(\sum_{i=1}^m \mathbf{\Lambda}_i \mathbf{1}\{\boldsymbol{\phi} \in \mathcal{X}_i\}\right)^{-1} \boldsymbol{\phi}
\end{aligned}$$

$$\leq 2\gamma^2\epsilon^{-1}$$

where the last inequality comes from Theorem 3 and the second inequality holds since, writing $\boldsymbol{\phi} = a\boldsymbol{u}_j + b\boldsymbol{v}$ for $\boldsymbol{v}^\top \boldsymbol{u}_j = 0$, we have

$$\boldsymbol{\phi}^\top \left(\sum_{i=1}^m \boldsymbol{\Lambda}_i\right)^{-1} \boldsymbol{\phi} = a^2 \boldsymbol{u}_j^\top \left(\sum_{i=1}^m \boldsymbol{\Lambda}_i\right)^{-1} \boldsymbol{u}_j + \underbrace{b^2 \boldsymbol{v}^\top \left(\sum_{i=1}^m \boldsymbol{\Lambda}_i\right)^{-1} \boldsymbol{v}}_{\geq 0} + \underbrace{2ab\boldsymbol{u}_j^\top \left(\sum_{i=1}^m \boldsymbol{\Lambda}_i\right)^{-1} \boldsymbol{v}}_{=0}.$$

Step 3: Concluding the Proof. Now we are ready for final proof. Using what we have just shown, we have for any $\boldsymbol{u}_j$:

$$\boldsymbol{u}_j^\top \left(\sum_{i=1}^m \sum_{(s_{h,\tau}, a_{h,\tau}) \in \mathcal{D}_i} \boldsymbol{\phi}(s_{h,\tau}, a_{h,\tau}) \boldsymbol{\phi}(s_{h,\tau}, a_{h,\tau})^\top\right) \boldsymbol{u}_j = \boldsymbol{u}_j^\top \left(\sum_{i=1}^m \boldsymbol{\Lambda}_i\right) \boldsymbol{u}_j - m\lambda \geq \left(\frac{\epsilon}{2\gamma^2} - \lambda m\right) \geq \frac{\epsilon}{4\gamma^2}.$$

The last inequality holds since $\lambda = \min\{1, \frac{\epsilon}{4m\gamma^2}\}$. Now notice that from Corollary 2, we have total sample complexity at most

$$\widetilde{\mathcal{O}}\left(\sum_{i=1}^m 2^i \cdot \max\left\{\frac{d}{\gamma_i^2} \log \frac{2^i/\lambda}{\gamma_i^2}, d^4 H^3 m^3 \log^3 \frac{1}{\delta}\right\}\right).$$

Using $m = \log(2/\epsilon)$ and $\lambda = \min\{1, \frac{\epsilon}{4m\gamma^2}\}$, we have the complexity upper bound for any h of

$$\widetilde{\mathcal{O}}\left(\frac{1}{\epsilon} \cdot \max\left\{\frac{d}{\gamma^2}, d^4 H^3 \log^3 \frac{1}{\delta}\right\}\right).$$

□